Hybrid Logics
Undecidability and infinite models

Carlos Areces
carlos.areces@gmailcom

Spring Term 2018
Stanford

What we cover in this lecture

- Today we will see some examples of modal logics which are undecidable.
- We will show some techniques used to prove undecidability, and other related properties.

The logic $\mathcal{HL}(\downarrow)$

- We already discussed it would be interesting to have dynamic naming, or "variables" in addition to nominals.
- Suposse we can create names "on the fly". Introduce the $\downarrow x$, that names the current state x.
- $\downarrow x$ names the current evaluation point, and let us refer to it in the rest of the formula. E.g., $\downarrow x.\Diamond x$ characterizes reflexive points.

The logic $\mathcal{HL}(\downarrow)$

- We already discussed it would be interesting to have dynamic naming, or "variables" in addition to nominals.
- Suposse we can create names "on the fly". Introduce the $\downarrow x$, that names the current state x.
- $\downarrow x$ names the current evaluation point, and let us refer to it in the rest of the formula. E.g., $\downarrow x.\Diamond x$ characterizes reflexive points.

To the signature of the basic modal logic add an infinite enumerable set of variables $\textsc{var}$.

Syntax of $\mathcal{HL}(\downarrow)$

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

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

The logic $\mathcal{HL}(\downarrow)$

The semantics of this logics is defined over usual Kripke models $\mathcal{M} = \langle W, \{R_i\}, V \rangle$ plus a assignment function $g : \textsc{var} \to W$ which assigns variables to elements in the domain.

The logic $\mathcal{HL}(\downarrow)$

The semantics of this logics is defined over usual Kripke models $\mathcal{M} = \langle W, \{R_i\}, V\rangle$ plus a assignment function $g : \textsc{var} \to W$ which assigns variables to elements in the domain.

- Given a model $\mathcal{M} = \langle W, \{R_i\}, V\rangle$ and an assignament g define:

$\mathcal{HL}(\downarrow)$ Semantics:

$$
\begin{array}{rcl}
\mathcal{M}, g, w \models x & \text{iff} & g(x) = w \\
\mathcal{M}, g, w \models \downarrow x.\varphi & \text{iff} & \mathcal{M}, g^x_w, w \models \varphi \text{ where } g^x_w \text{ is identical to } g \\
 & & \text{except } g^x_w(x) = w.
\end{array}
$$

Infinite model

- We saw that the basic modal logic (and other extensions) have the finite model property
- This is useful for proving decidability (knowing also a bound for the model size)
- We will see that $\mathcal{HL}(\downarrow)$ is able to force an infinite model
- By itself, this does not prove undecidability, but it paves the way

Infinite model

- Write a formula which says that there is a non empty set B whose elements constitute a strict partial order. That is:
  - Irreflexive
  - Transitive
- And where every element has a successor

This implies that B is infinite.

Infinite model

- Write a formula which says that there is a non empty set B whose elements constitute a strict partial order. That is:
  - Irreflexive
  - Transitive
- And where every element has a successor

This implies that B is infinite.

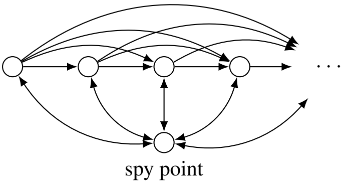
- How can we speak about a set of points with a modal formula that is evaluated in a single point? Spy Point!

Infinite model

- Write a formula which says that there is a non empty set B whose elements constitute a strict partial order. That is:
  - Irreflexive
  - Transitive
- And where every element has a successor

This implies that B is infinite.

- How can we speak about a set of points with a modal formula that is evaluated in a single point? Spy Point!



Forcing and infinite model

Each s-successor has an edge towards s

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

Forcing and infinite model

Each s-successor has an edge towards s

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

The s-successors in two steps are s-successors in one step

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

Forcing and infinite model

Each s-successor has an edge towards s

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

The s-successors in two steps are s-successors in one step

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

The relation over the s-successors is irreflexive

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

Forcing and infinite model

Each s-successor has an edge towards s

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

The s-successors in two steps are s-successors in one step

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

The relation over the s-successors is irreflexive

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

Every s-successors has a successor which is not s

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

Forcing and infinite model

Each s-successor has an edge towards s

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

The s-successors in two steps are s-successors in one step

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

The relation over the s-successors is irreflexive

(*Irr*) $[r]\neg(\downarrow x.\langle r\rangle x)$

Every s-successors has a successor which is not s

(*Succ*) $\downarrow s.([r]\langle r\rangle\neg s)$

The relation over the s-successors is transitive

(*Tran*) $\downarrow s.[r]\downarrow x.[r](\neg s \rightarrow [r](\neg s \rightarrow \downarrow z.\langle r\rangle(s \wedge \langle r\rangle(x \wedge \langle r\rangle z))))$

Forcing and infinite model

Each s-successor has an edge towards s

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

The s-successors in two steps are s-successors in one step

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

The relation over the s-successors is irreflexive

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

Every s-successors has a successor which is not s

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

The relation over the s-successors is transitive

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

Let φ be the formula $\mathit{Back} \wedge \mathit{Spy} \wedge \mathit{Irr} \wedge \mathit{Succ} \wedge \mathit{Tran}$

Forcing an infinite model

Theorem

If $\mathcal{M}, g, w \models \varphi$ then $\mathcal{M}$ is infinite

Proof. By construction of φ.

Forcing an infinite model

Theorem

If $\mathcal{M}, g, w \models \varphi$ then $\mathcal{M}$ is infinite

Proof. By construction of φ.

We have to check also that φ does have models.

Theorem

There exists $\mathcal{M}, g, w$ s.t. $\mathcal{M}, g, w \models \varphi$.

Proof. Let B be an infinite set of elements and w an element such that $w \notin B$. Let R be the smallest relation such that

- R defines an strict partial order over B
- wRb and bRw for every element $b \in B$

$\mathcal{M} = \langle B \cup \{w\}, R, V\rangle$ verifies $\mathcal{M}, g, w \models \varphi$ (for any V and g).

Undecidability

- How do we prove that a logic is undecidable?
- If we want to prove it in a direct way, we must write a formula that codifies arbitrary executions in a Turing machine
- The tiling problem, which has been proved to be undecidable, will be useful for the modal case

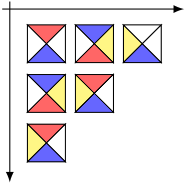
The tiling problem

- Given a finite set of types of tiles $\mathcal{T}$

The tiling problem

- Given a finite set of types of tiles $\mathcal{T}$

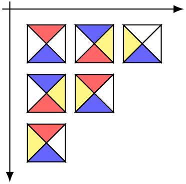
The tiling problem: Is it possible to put tiles of type $\mathcal{T}$ in $\mathbb{N} \times \mathbb{N}$ such that each pair of adjacent tiles has the same color?



The tiling problem

- Given a finite set of types of tiles $\mathcal{T}$

The tiling problem: Is it possible to put tiles of type $\mathcal{T}$ in $\mathbb{N} \times \mathbb{N}$ such that each pair of adjacent tiles has the same color?

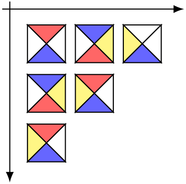


- The tiling problem in $\mathbb{N} \times \mathbb{N}$ is known to be undecidable

The tiling problem

- Given a finite set of types of tiles $\mathcal{T}$

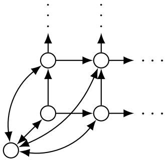
The tiling problem: Is it possible to put tiles of type $\mathcal{T}$ in $\mathbb{N} \times \mathbb{N}$ such that each pair of adjacent tiles has the same color?



- The tiling problem in $\mathbb{N} \times \mathbb{N}$ is known to be undecidable
- Given a set of types of tiles $\mathcal{T}$, we want to write a formula $\varphi_{\mathcal{T}}$ such that $\varphi_{\mathcal{T}}$ is satisfiable iff there exists a tiling for $\mathcal{T}$

The tiling problem for $\mathcal{HL}(\downarrow)$

- We will use again a spy point



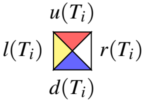
- (Notice, codifying the tiling problem does not imply forcing an infinite model, why?)

The tiling problem for $\mathcal{HL}(\downarrow)$

- Let $T = \{T_1, \ldots, T_n\}$ be a set of types of tiles

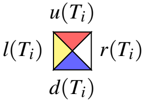
The tiling problem for $\mathcal{HL}(\downarrow)$

- Let $T = \{T_1, \ldots, T_n\}$ be a set of types of tiles
- Given a type of tile T_i, $u(T_i)$, $r(T_i)$, $d(T_i)$, $l(T_i)$ represent the colors of T_i corresponding to their sides.

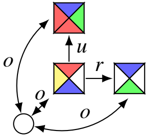


The tiling problem for $\mathcal{HL}(\downarrow)$

- Let $T = \{T_1, \ldots, T_n\}$ be a set of types of tiles
- Given a type of tile T_i, $u(T_i)$, $r(T_i)$, $d(T_i)$, $l(T_i)$ represent the colors of T_i corresponding to their sides.



- Suppose that we also have a modality $\langle o \rangle$ that we use to move from the spy point into every tile
- And we have as well modalities $\langle u \rangle$ and $\langle r \rangle$ in order to move up from a tile and to the right of a tile respectively.



The tiling problem for $\mathcal{HL}(\downarrow)$

First we codify the grid:

Each s-successor has an edge towards s

$(\textit{Back}) \qquad [o]\neg s \wedge \langle o \rangle \top \wedge [o]\langle o \rangle s$

The tiling problem for $\mathcal{HL}(\downarrow)$

First we codify the grid:

Each s-successor has an edge towards s

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

The successor of a tile is an s-successor

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

The tiling problem for $\mathcal{HL}(\downarrow)$

First we codify the grid:

Each s-successor has an edge towards s

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

The successor of a tile is an s-successor

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

From a tile, s cannot be reached using r and u

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

The tiling problem for $\mathcal{HL}(\downarrow)$

First we codify the grid:

Each s-successor has an edge towards s

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

The successor of a tile is an s-successor

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

From a tile, s cannot be reached using r and u

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

Every tiles has a tile above and to the right

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

The tiling problem for $\mathcal{HL}(\downarrow)$

First we codify the grid:

Each s-successor has an edge towards s

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

The successor of a tile is an s-successor

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

From a tile, s cannot be reached using r and u

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

Every tiles has a tile above and to the right

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

Each tile has only one tile above and only one tile to the right

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

The tiling problem for $\mathcal{HL}(\downarrow)$

First we codify the grid:

Each s-successor has an edge towards s

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

The successor of a tile is an s-successor

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

From a tile, s cannot be reached using r and u

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

Every tiles has a tile above and to the right

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

Each tile has only one tile above and only one tile to the right

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

There is concluence between up-right and right-up

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

The tiling problem for $\mathcal{HL}(\downarrow)$

Finally, there is a tile in each point of the grid and all the colors coincide:

Each point has a single type of tile

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

The tiling problem for $\mathcal{HL}(\downarrow)$

Finally, there is a tile in each point of the grid and all the colors coincide:

Each point has a single type of tile

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

Each tile has an adjacent tile above which is appropriately colored

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

The tiling problem for $\mathcal{HL}(\downarrow)$

Finally, there is a tile in each point of the grid and all the colors coincide:

Each point has a single type of tile

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

Each tile has an adjacent tile above which is appropriately colored

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

Each tile has an adjacent tile to the right which is appropriately colored

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

The tiling problem for $\mathcal{HL}(\downarrow)$

Finally, there is a tile in each point of the grid and all the colors coincide:

Each point has a single type of tile

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

Each tile has an adjacent tile above which is appropriately colored

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

Each tile has an adjacent tile to the right which is appropriately colored

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

Let

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

The tiling problem for $\mathcal{HL}(\downarrow)$

Theorem

Let T be a set of tile types. Then φ_T is satisfiable iff there is a T-tiling of $\mathbb{N} \times \mathbb{N}$.

The tiling problem for $\mathcal{HL}(\downarrow)$

Theorem

Let T be a set of tile types. Then φ_T is satisfiable iff there is a T-tiling of $\mathbb{N} \times \mathbb{N}$.

Proof:

($\Rightarrow$) Suppose that $\mathcal{M}, w \models \varphi_T$. By construction, $\mathcal{M}$ represents a tiling in $\mathbb{N} \times \mathbb{N}$.

The tiling problem for $\mathcal{HL}(\downarrow)$

Theorem

Let T be a set of tile types. Then φ_T is satisfiable iff there is a T-tiling of $\mathbb{N} \times \mathbb{N}$.

Proof:

($\Rightarrow$) Suppose that $\mathcal{M}, w \models \varphi_T$. By construction, $\mathcal{M}$ represents a tiling in $\mathbb{N} \times \mathbb{N}$.

($\Leftarrow$) Suppose that $f : \mathbb{N} \times \mathbb{N} \to T$ is a tiling in $\mathbb{N} \times \mathbb{N}$. We define the model $\mathcal{M} = \langle W, \{R_o, R_u, R_r\}, V\rangle$:

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

It is not hard to see that $\mathcal{M}, w \models \varphi_T$

A tiling for every need

- Tiling problem are not only useful for proving undecidability

A tiling for every need

- Tiling problem are not only useful for proving undecidability
- Tiling problems are also useful for proving complexity bounds and other degrees of undecidability. For example:

A tiling for every need

- Tiling problem are not only useful for proving undecidability
- Tiling problems are also useful for proving complexity bounds and other degrees of undecidability. For example:
  - The tiling problem we just saw is Π^0_1-complete.

A tiling for every need

- Tiling problem are not only useful for proving undecidability
- Tiling problems are also useful for proving complexity bounds and other degrees of undecidability. For example:
  - The tiling problem we just saw is Π^0_1-complete.
  - If we distinguish T_1, a particular type of tile, the tiling problem in $\mathbb{N} \times \mathbb{N}$ where T_1 occurs infinitely often in the first row is Σ^1_1-complete.

A tiling for every need

- Tiling problem are not only useful for proving undecidability
- Tiling problems are also useful for proving complexity bounds and other degrees of undecidability. For example:
  - The tiling problem we just saw is Π^0_1-complete.
  - If we distinguish T_1, a particular type of tile, the tiling problem in $\mathbb{N} \times \mathbb{N}$ where T_1 occurs infinitely often in the first row is Σ^1_1-complete.
  - The “two person corridor tiling” is EXPTIME-complete

Two person corridor tiling

- This tiling has 3 players: Spoiler, Duplicator and a referee.

Two person corridor tiling

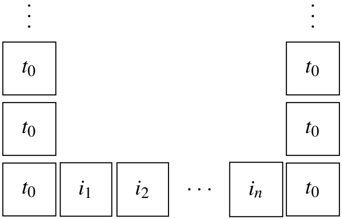
- This tiling has 3 players: Spoiler, Duplicator and a referee.
- From the finite set of types of tiles $T = \{t_0, t_1, \ldots, t_{s+1}\}$ we will distinguish t_0 y t_{s+1}

Two person corridor tiling

- This tiling has 3 players: Spoiler, Duplicator and a referee.
- From the finite set of types of tiles $T = \{t_0, t_1, \ldots, t_{s+1}\}$ we will distinguish t_0 y t_{s+1}
- As an input parameter we also get a $n \in \mathbb{N}$, which defines the width of the corridor

Two person corridor tiling

- This tiling has 3 players: Spoiler, Duplicator and a referee.
- From the finite set of types of tiles $T = \{t_0, t_1, \ldots, t_{s+1}\}$ we will distinguish t_0 y t_{s+1}
- As an input parameter we also get a $n \in \mathbb{N}$, which defines the width of the corridor
- The game starts with the referee putting tiles as follows



Two person corridor tiling

- The rules for completing the corridor are strict: from down to up and from left to right

Two person corridor tiling

- The rules for completing the corridor are strict: from down to up and from left to right
- I.e., the players cannot choose where to put a tile, they can only choose the type of tile

Two person corridor tiling

- The rules for completing the corridor are strict: from down to up and from left to right
- I.e., the players cannot choose where to put a tile, they can only choose the type of tile
- The rules about color coincidence are the usual ones (and they include the “borders” of the corridor)

Two person corridor tiling

- The rules for completing the corridor are strict: from down to up and from left to right
- I.e., the players cannot choose where to put a tile, they can only choose the type of tile
- The rules about color coincidence are the usual ones (and they include the “borders” of the corridor)
- The players play alternatively, and Spoiler plays first.

Two person corridor tiling

- The rules for completing the corridor are strict: from down to up and from left to right
- I.e., the players cannot choose where to put a tile, they can only choose the type of tile
- The rules about color coincidence are the usual ones (and they include the “borders” of the corridor)
- The players play alternatively, and Spoiler plays first.
- When do the players win or lose?

Two person corridor tiling

- The rules for completing the corridor are strict: from down to up and from left to right
- I.e., the players cannot choose where to put a tile, they can only choose the type of tile
- The rules about color coincidence are the usual ones (and they include the "borders" of the corridor)
- The players play alternatively, and Spoiler plays first.
- When do the players win or lose?
  - If after a finite number of rounds a tile of type t_{s+1} is put in the first column, Spoiler wins

Two person corridor tiling

- The rules for completing the corridor are strict: from down to up and from left to right
- I.e., the players cannot choose where to put a tile, they can only choose the type of tile
- The rules about color coincidence are the usual ones (and they include the “borders” of the corridor)
- The players play alternatively, and Spoiler plays first.
- When do the players win or lose?
  - If after a finite number of rounds a tile of type t_{s+1} is put in the first column, Spoiler wins
  - Otherwise (if no player can make a valid move, t_{s+1} is not in the column 1, or the game goes on infinitely) Duplicator wins

Two person corridor tiling

- The rules for completing the corridor are strict: from down to up and from left to right
- I.e., the players cannot choose where to put a tile, they can only choose the type of tile
- The rules about color coincidence are the usual ones (and they include the “borders” of the corridor)
- The players play alternatively, and Spoiler plays first.
- When do the players win or lose?
  - If after a finite number of rounds a tile of type t_{s+1} is put in the first column, Spoiler wins
  - Otherwise (if no player can make a valid move, t_{s+1} is not in the column 1, or the game goes on infinitely) Duplicator wins
- The problem of detecting if Spoiler has a winning strategy is known to be EXPTIME-complete.

Two person corridor tiling

- Using the “two person corridor tiling” one can prove that PDL is EXPTIME-hard, codifying the tree of possible moves between Spoiler y Duplicator.
- Also, it can be used to prove that $K + A$ is EXPTIME-hard.