4. Complexity of selected DLs

So far: upper complexity bounds for the satisfiability of:

- **CSAT($\mathcal{ALC}$) is in** NExpTime
  (tableau algorithm is non-deterministic, builds c-tree of linear depth)

- **SAT($\mathcal{ALCQI}$) w.r.t. TBoxes is in** 2NExpTime
  (tableau algorithm is non-deterministic, builds c-tree of exponential depth)

- **SAT($\mathcal{ALCIO}$) w.r.t. TBoxes is in** ExpTime
  (automata built for $C_0, \mathcal{T}$ is exponential in $|C_0| + |\mathcal{T}|$,
  testing its emptiness is polynomial in automata's size)

PSpace DLs: $\mathcal{ALC}$ is in PSpace

The **NExpTime** tableau algorithm deciding **CSAT**$(\mathcal{ALC})$ can be modified easily to run in **PSpace**:

For an $\mathcal{ALC}$-concept C_0,

1. the c-tree can be built **depth-first**
2. **branches are independent** $\leadsto$ keep only one branch in memory at any time
3. **length** of branch $\leq |C_0|$
4. for each node x, $\mathfrak{L}(x) \subseteq \mathsf{sub}(C_0)$ and $\#\,\mathsf{sub}(C_0)$ is linear in $|C_0|$

$\leadsto$ non-deterministic PSpace decision procedure for **CSAT**$(\mathcal{ALC})$

inverse roles: use "R-neighbour" instead of "R-successor"
to decide **CSAT**$(\mathcal{ALCI})$ $\rightsquigarrow$ tableau algorithm satisfying

1. the c-tree can be built depth-first
2. branches are independent $\rightsquigarrow$ keep only one branch in memory at any time
3. **length** of branch $\leq |C_0|$
4. for each node x, $\mathfrak{L}(x) \subseteq$ **sub**(C_0) and $\#$ **sub**(C_0) is linear in $|C_0|$

branches are not independent: effects go **up one branch, down another branch**

Solution: **reset-restart** technique:

5. when node label $\mathfrak{L}(x)$ changes, remove the subtree below x (reset) and re-construct it (restart).

$\rightsquigarrow$ **PSpace** realisation of the tableau algorithm for **CSAT**$(\mathcal{ALCI})$

PSpace DLs: $\mathcal{ALC}$ (and thus $\mathcal{ALCI}$) is PSpace-hard

PSpace-hardness of $\mathcal{ALC}$ proved by a reduction of

the validity problem for quantified Boolean formulae to **satisfiability of $\mathcal{ALC}$**

QBFs are of the form $Q_1 p_1 . Q_2 p_2 . \ldots . Q_n p_n . \varphi$
for φ a Boolean formula over $p_1, \ldots, p_n$ and $Q_i \in \{\forall, \exists\}$.

Validity of QBFs defined inductively:

$\exists p . \Phi$ is valid if $\Phi[p/t]$ or $\Phi[p/f]$ is valid
$\forall p . \Phi$ is valid if $\Phi[p/t]$ and $\Phi[p/f]$ are valid

Known: validity of QBFs is PSpace-hard

Obligation: for QBF Φ, define C_Φ such that Lemma: Φ is valid iff C_Φ is satisfiable

⇝ Corollary: **CSAT**($\mathcal{ALC}$) is PSpace-complete

Each model of $C_\Phi = Q_1 p_1.Q_2 p_2.\ldots Q_n p_n.\varphi$ contains a tree of depth n where, for $x_0 \in C_\Phi$,

- if $Q_i = \exists$, then each $R^{(i-1)}$-successor of x_0 has 1 R-successor, and
- if $Q_i = \forall$, then each $R^{(i-1)}$-successor of x_0 has 2 R-successors, one in p_i and one in $\neg p_i$.
- and all leave nodes satisfy φ.

$$C_\Phi := L_1 \sqcap \forall R.(L_2 \sqcap \forall R.(L_3 \sqcap \ldots \forall R.(L_n \sqcap \varphi)) \ldots)) \text{ where}$$

$$L_i := D_i \sqcap \begin{cases} \exists R.\top & \text{if } Q_i = \exists \\ \exists R.p_i \sqcap \exists R.\neg p_i & \text{if } Q_i = \forall \end{cases} \text{ and}$$

$$D_i := \bigsqcap_{j \leq i} (p_j \Rightarrow \forall R.p_j) \sqcap (\neg p_j \Rightarrow \forall R.\neg p_j)$$

□

ExpTime DLs: $\mathcal{ALC}$ with TBoxes is ExpTime-hard

We know: **SAT**($\mathcal{ALCIO}$) w.r.t. TBoxes is in **ExpTime**

Lemma: **SAT**($\mathcal{ALC}$) w.r.t. TBoxes is **ExpTime-hard**

Proof idea: reduce the halting problem of a **polynomial-space-bounded alternating TM** to **SAT**($\mathcal{ALC}$) w.r.t. TBoxes.

Basic Ideas: for PSB-A-TM $\mathcal{M}$, build TBox $\mathcal{T}$ with

- **tape-cell i contains letter σ** is coded as concept name σ_i (possible due to polynomial space bound)
- use CEs to ensure that each tape-cell contains exactly one letter
- use CEs to encode $\mathcal{M}$'s ($\forall/\exists$) transitions relation (e.g., cell contents changes only near head)

Then $\mathcal{M}$ **holds on** w iff $\hat{w}$ is satisfiable w.r.t. $\mathcal{T}$

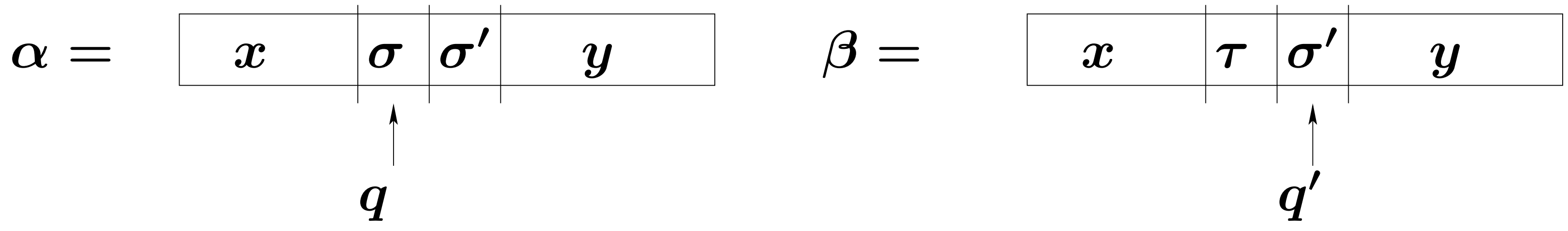
An **alternating TM** $\mathcal{M} = (Q, \Sigma, \Delta, q_0)$ consists of

$$
\begin{aligned}
Q = U \,\dot\cup\, E, &\quad \text{a set of **states** with} \\
U &\quad \text{a set of **universal states** and} \\
E &\quad \text{a set of **existential states**,} \\
q_0 \in Q &\quad \text{the **initial state**,} \\
\Sigma, &\quad \text{the **alphabet**,} \\
\Delta \subseteq (Q \times \Sigma) \times (Q \times \Sigma \times \{\ell, r\}), &\quad \text{the **transition relation**.}
\end{aligned}
$$

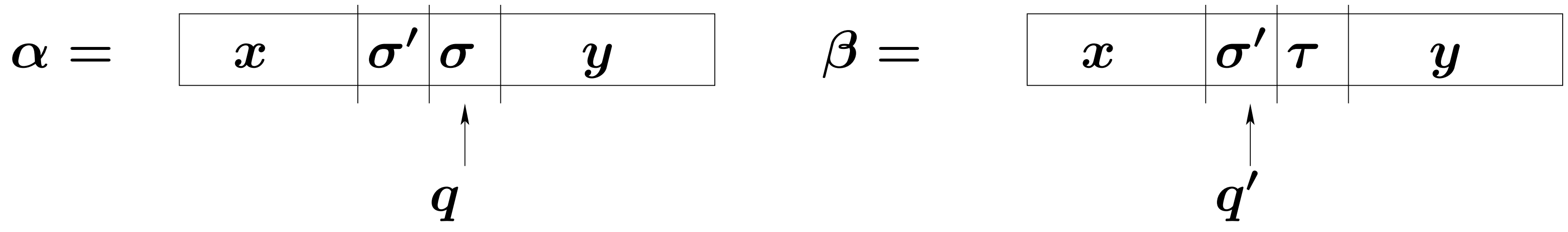
- $\alpha \in \Sigma^* Q \Sigma^+$ is a **configuration**,
- $\alpha \in \Sigma^* U \Sigma^+$ is a **universal configuration**,
- $\alpha \in \Sigma^* E \Sigma^+$ is an **existential configuration**.

The transitions are defined as usual:

β is a next-config of α
if $(q, \sigma, q', \tau, r) \in \Delta$ and



or
if $(q, \sigma, q', \tau, l) \in \Delta$ and



A **trace** of $\mathcal{M}$ is a finite set $T = \{\alpha_0, \alpha_1, \ldots\}$ where each α_i is a configuration and if

- $\alpha \in T$ is a universal configuration, then $\forall$ next configuration β of α: $\beta \in T$
- $\alpha \in T$ is an existential configuration, then $\exists$ a next configuration β of α: $\beta \in T$.

$$L(\mathcal{M}) := \{w \in \Sigma^* \mid \exists T \text{ with } q_0 w \in T\}$$

$s(\cdot)$ – bounded A-TM:

$$L(\mathcal{M}, s(\cdot)) := \{w \in \Sigma^* \mid \exists T \text{ with } q_0 w \in T \text{ and } \forall \alpha \in T : |\alpha| \leq s(|w|)\}$$

ExpTime DLs: $\mathcal{ALC}$ with TBoxes is ExpTime-hard

For an ATM $\mathcal{M} = (Q, \Sigma, b, \Delta, q_0)$, a word $w \in \Sigma^*$ and $m := s(|w|) + 1$, the TBox $\mathcal{T}_\mathcal{M}$ contains the following CEs to ensure that each element in a model corresponds to a configuration of $\mathcal{M}$:

exactly one state and one head position

$$\top \doteq \bigsqcup_{q \in Q} (A_q \sqcap \bigsqcap_{q' \neq Q} (\neg A_{q'})) \sqcap \bigsqcup_{i=0}^{m} (H_i \sqcap \bigsqcap_{j \neq i} \neg H_j)$$

exactly one letter per cell

$$\top \doteq \bigsqcap_{i=0}^{m} \bigsqcup_{\sigma \in \Sigma} (C_{i,\sigma} \sqcap \bigsqcap_{\tau \neq \sigma} \neg C_{i,\tau})$$

unread cells are maintained

$$\top \doteq \bigsqcap_{i=0}^{m} \bigsqcap_{\sigma \in \Sigma} (C_{i,\sigma} \sqcap \neg H_i \Rightarrow \forall N.C_{i,\sigma})$$

head moves only 1 cell

$$\top \doteq \bigsqcap_{i=0}^{m} (H_i \Rightarrow \forall N.(H_{i-1} \sqcup H_{i+1}))$$

ExpTime DLs: $\mathcal{ALC}$ with TBoxes is ExpTime-hard

universal transitions are ok

$$\top \doteq \bigsqcap_{i=1}^{m-1} \bigsqcap_{\sigma,q \in U} H_i \sqcap C_{i,\sigma} \sqcap A_q \Rightarrow \bigsqcap_{(q,\sigma,q',\sigma',r)\in\Delta} \exists N.(H_{i+1} \sqcap C_{i,\sigma'} \sqcap A_{q'}) \sqcap \bigsqcap_{(q,\sigma,q',\sigma',l)\in\Delta} \exists N.(H_{i-1} \sqcap C_{i,\sigma'} \sqcap A_{q'})$$

existential transitions are ok

$$\top \doteq \bigsqcap_{i=1}^{m-1} \bigsqcap_{\sigma,q \in E} H_i \sqcap C_{i,\sigma} \sqcap A_q \Rightarrow \bigsqcup_{(q,\sigma,q',\sigma',r)\in\Delta} \exists N.(H_{i+1} \sqcap C_{i,\sigma'} \sqcap A_{q'}) \sqcup \bigsqcup_{(q,\sigma,q',\sigma',l)\in\Delta} \exists N.(H_{i-1} \sqcap C_{i,\sigma'} \sqcap A_{q'})$$

Then, for $w = \sigma_1 \cdots \sigma_n$, we have that

$$A_{q_0} \sqcap H_1 \sqcap \bigsqcap_{i=1}^{n} C_{i,\sigma_i}$$

is satisfiable w.r.t. $\mathcal{T}_{\mathcal{M}}$ iff $w \in L(\mathcal{M})$.

Each Description Logic extending $\mathcal{ALC}$ where we can

polynomially reduce satisfiability w.r.t. TBoxes to
pure concept satisfiability is
ExpTime-hard

This reduction is called **internalisation** of TBoxes

ExpTime DLs: $\mathcal{ALCIO}$ is ExpTime-hard, even without TBoxes

Lemma: **CSAT($\mathcal{ALCIO}$) is ExpTime-hard**

Proof idea: use a **spy to internalise TBox**

Spy: an object N related to **all** other objects via (new) role U:
for input concept C, write

$$N \sqcap \exists U.C \sqcap \forall U. \bigsqcap_{\text{role } R} \forall R.\exists U^{-}.N$$

where the conjunction is over all roles R, R^{-} in C or $\mathcal{T}$

Internalise: ensure that each object satisfies C iff it satisfies D
for each CE $C \doteq D$ in TBox; add:

$$\ldots \sqcap \forall U. \bigsqcap_{\ldots}(C \Leftrightarrow D)$$

$\rightsquigarrow$ **SAT($\mathcal{ALCIO}$) w.r.t. TBoxes is polyn. reducible to CSAT($\mathcal{ALCIO}$)**

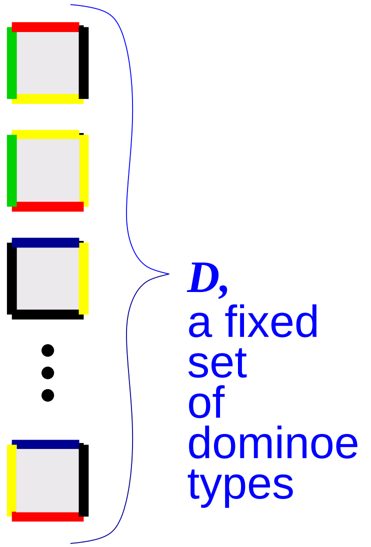
NExpTime DLs: $\mathcal{ALCQIO}$ is NExpTime-hard

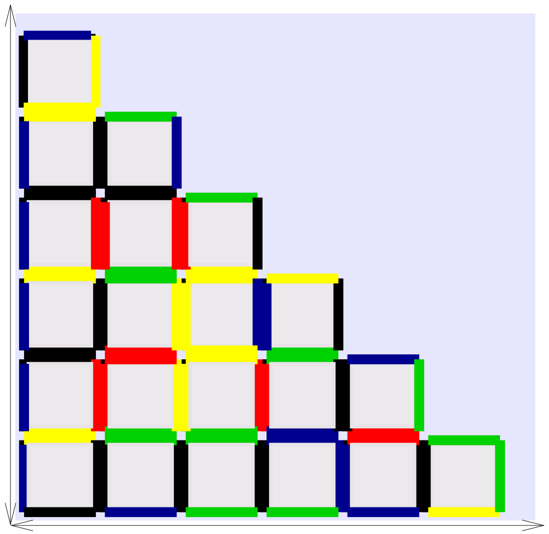
We know: **SAT**($\mathcal{ALCIO}$) w.r.t. TBoxes is **ExpTime-complete**

Also: **SAT**($\mathcal{ALCQI}$) w.r.t. TBoxes is **ExpTime-complete**
where $\mathcal{Q}$ stands for **qualifying number restrictions** $(\geqslant nr.C)$ and $(\leqslant nr.C)$

Lemma: their combination is **NExpTime-hard**

Proof: by reduction of a NExpTime version of the **domino problem:**





can we tile a 2^n x 2^n square using D?

Definition: A **domino system** $\mathcal{D} = (D, H, V)$

- set of domino **types** $D = \{D_1, \ldots, D_d\}$, and
- horizontal and vertical **matching conditions** $H \subseteq D \times D$ and $V \subseteq D \times D$

A **tiling** of the $\mathbb{N} \times \mathbb{N}$ grid using $\mathcal{D}$:

$$\begin{aligned} t : \mathbb{N} \times \mathbb{N} \longrightarrow\ & D \text{ such that} \\ & \langle t(m,n), t(m+1,n)\rangle \in H \text{ and} \\ & \langle t(m,n), t(m,n+1)\rangle \in V \end{aligned}$$

Domino problem

standard: has $\mathcal{D}$ a tiling?
NExpTime: has $\mathcal{D}$ a tiling for a $2^n \times 2^n$ square?

Reducing the NExpTime domino problem to **CSAT**($\mathcal{ALCQIO}$) $\rightsquigarrow$ **four tasks:**

① each object carries exactly one domino type D_i
$\rightsquigarrow$ use concept name D_i for each domino type and

$$\top \doteq \bigsqcup_{1 \leq i \leq d} (D_i \sqcap \bigsqcap_{j \neq i} \neg D_j)$$

② each element x has exactly one H-successor
exactly one V-successor
whose domino types satisfy the horizontal/vertical matching conditions:

$$\top \doteq \bigsqcap_{1 \leq i \leq n} D_i \Rightarrow \; ((\leqslant 1 V.\top) \sqcap (\exists V. \bigsqcup_{(D_i, D_j) \in V} D_j) \sqcap$$
$$((\leqslant 1 H.\top) \sqcap (\exists H. \bigsqcup_{(D_i, D_j) \in H} D_j)$$

③ the model must be large enough, i.e., have $2^n \times 2^n$ elements
⇝ encode the position (x, y) of each point using binary coding in the concept names $X_1, \ldots, X_n$, $Y_1, \ldots, Y_n$:

$$\top \doteq (\tilde{X}_1 \sqcap \tilde{X}_2 \sqcap \ldots \sqcap \tilde{X}_n)$$

$$\tilde{X}_1 \doteq (X_1 \sqcap (\forall H.\neg X_1)) \sqcup (\neg X_1 \sqcap (\forall H.X_1)) \qquad \text{\% switch lowest bit}$$

$$\tilde{X}_i \doteq (\bigsqcap_{1 \leq j < i} X_j \sqcap ((X_i \sqcap \forall H.\neg X_i) \sqcup (\neg X_i \sqcap \forall H.X_i))) \sqcup \quad \text{\% switch } i\text{-th bit, if}$$

$$\text{\% all lower ones are 1}$$

$$(\neg \bigsqcap_{1 \leq j < i} X_j \sqcap ((X_i \sqcap \forall H.X_i) \sqcup (\neg X_i \sqcap \forall H.\neg X_i))). \quad \text{\% else keep it}$$

E.g., an instance of $\neg X_3$, X_2, X_1, Y_3, $\neg Y_2$, and Y_1 represents the tuple $(011, 101)$, and thus the point $(3, 5)$.

④ ensure that the $V \circ H$-successor of each node coincides with its $H \circ V$-successor

$\rightsquigarrow$ enforce that each object is the H-successor of at most one element (and the same for V):

$$\top \doteq (\leqslant 1 V^{-}.\top) \sqcap (\leqslant 1 H^{-}.\top)$$

$\rightsquigarrow$ enforce that there is ≤ 1 object in the upper right corner:

$$N \doteq X_1 \sqcap \ldots \sqcap X_n \sqcap Y_1 \sqcap \ldots \sqcap Y_n$$

Harvest:

$$\neg X_1 \sqcap \ldots \sqcap \neg X_n \sqcap \neg Y_1 \sqcap \ldots \sqcap \neg Y_n$$

is satisfiable w.r.t. to $\mathcal{T}_D$ defined above iff D has a $2^n \times 2^n$-tiling.

NExpTime DLs: $\mathcal{ALCQIO}$ is in NExpTime

Known:

① 2-variable fragment of FOL is decidable

② 2-variable fragment of FOL **with counting quantifiers** is decidable
$\exists^{\geq n} x \ldots$ and $\exists^{\leq n} x \ldots$ Nexptime-complete

③ in $\mathcal{ALCIO}$, TBoxes can be internalized

$\rightsquigarrow$ to prove **SAT**($\mathcal{ALCQIO}$) w.r.t. TBoxes being in NExpTime, it suffices to reduce **CSAT**($\mathcal{ALCQIO}$) to satisfiability of C-FOL2

In $\mathcal{ALCQIO}$, $\exists R.C$ and $\forall R.C$ are syntactic sugar:

$$\exists R.C \equiv (\geqslant 1 R.C) \quad \text{and} \quad \forall R.C \equiv (\leqslant 0 R.\neg C)$$

NExpTime DLs: $\mathcal{ALCQIO}$ is in NExpTime

$$
\begin{aligned}
t_x(A) &= \mathrm{A}(x),\\
t_x(N) &= (x = N),\\
t_x(\neg D) &= \neg t_x(D),\\
t_x(C \sqcap D) &= t_x(C) \wedge t_x(D),\\
t_x(C \sqcup D) &= t_x(C) \vee t_x(D),\\
t_x(\geqslant n R.C) &= \exists^{\geq n} y.\mathrm{R}(x,y) \wedge t_y(C),\\
t_x(\geqslant n R^-.C) &= \exists^{\geq n} y.\mathrm{R}(y,x) \wedge t_y(C),\\
t_x(\leqslant n R.C) &= \exists^{\leq n} y.\mathrm{R}(x,y) \wedge t_y(C),\\
t_x(\leqslant n R^-.C) &= \exists^{\leq n} y.\mathrm{R}(y,x) \wedge t_y(C).
\end{aligned}
$$

$\rightsquigarrow$ **SAT**($\mathcal{ALCQIO}$) w.r.t. TBoxes is NExpTime-complete

SAT$(\mathcal{ALCQIO})$ w.r.t. TBoxes is NExpTime-complete

provided that $|(\geqslant n R.C)| = |(\leqslant n R.C)| = n + 1 + |C|$.

For $|(\geqslant n R.C)| = |(\leqslant n R.C)| = \log(n) + 1 + |C|$,
no exact upper bound is known.

The DL $\mathcal{SHIQ}$ extends $\mathcal{ALCQI}$ with **transitive roles** and **role hierarchies**:

transitive roles: certain role names must be interpreted as **transitive** relations
e.g., ancestor, has-part, part-of, etc.

role hierarchy: set of implications $R \sqsubseteq S$, which require $R^\mathcal{I} \subseteq S^\mathcal{I}$
e.g., daughter $\sqsubseteq$ child, has-component $\sqsubseteq$ has-part

FaCT and **Racer** are highly optimised $\mathcal{SHIQ}$-implementations

Note: If **Trans**(S) and $R \sqsubseteq S$, then $S^\mathcal{I}$ is **a** transitive relation containing $R^\mathcal{I}$

$$\neq$$

$(R^+)^\mathcal{I}$ is **the smallest** transitive relation containing $R^\mathcal{I}$

Known: **SAT**$(\mathcal{SHIQ})$ w.r.t. TBoxes is ExpTime-complete if
roles in NRs are **simple** (don't have transitive subroles)

$\mathcal{SHIQ}$ is undecidable with non-simple roles in NRs

Restricting NRs to simple roles is indeed necessary

Proof: $\mathcal{SHIQ}$ without inverse roles and with un-qualifiying NRs only, but with non-simple roles in NRs is undecidable

by **reduction of the domino problem:**

For a domino system $\mathcal{D} = (D, H, V)$, define a TBox $\mathcal{T}_\mathcal{D}$ and a $\mathcal{SHIQ}$-concept $C_\mathcal{D}$ such that

$\mathcal{D}$ has a tiling (of the whole grid) iff $C_\mathcal{D}$ is satisfiable w.r.t. $\mathcal{T}_\mathcal{D}$

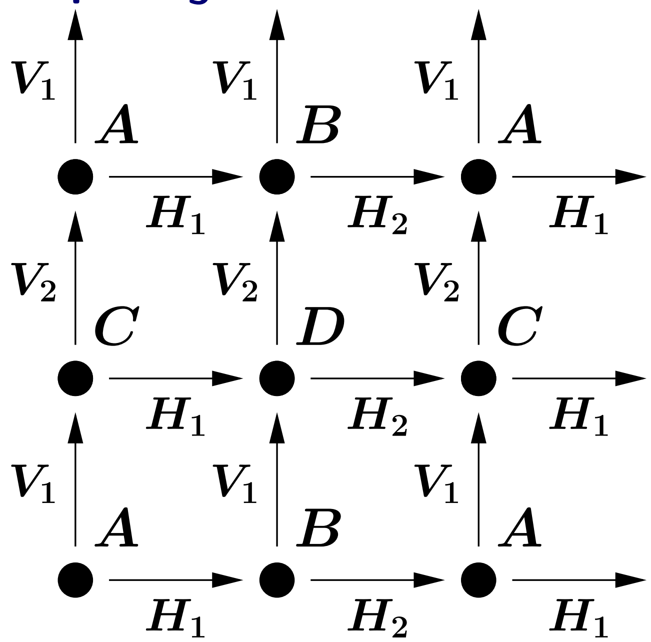
We can (almost) re-use the concept equations

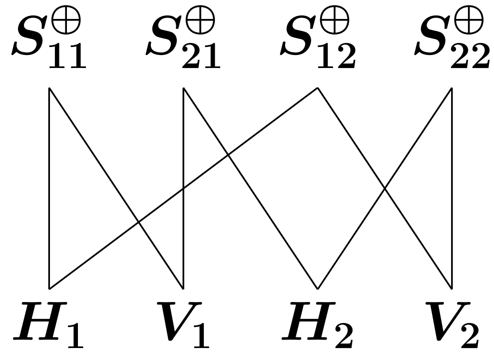
$$\top \doteq (\bigsqcup_{1 \leq i \leq n} D_i) \sqcap (\bigsqcap_{1 \leq i < j \leq n} \neg(D_i \sqcap D_j))$$

$$\top \doteq \bigsqcap_{1 \leq i \leq n} D_i \Rightarrow ((\leqslant 1 V.\top) \sqcap (\exists V. \bigsqcup_{(D_i, D_j) \in V} D_j) \sqcap$$

$$\bigsqcap_{1 \leq i \leq n} D_i \Rightarrow ((\leqslant 1 H.\top) \sqcap (\exists H. \bigsqcup_{(D_i, D_j) \in H} D_j)))$$

… and impose grid-structure as follows:





$$\begin{aligned}
\top &\doteq A \Rightarrow (\neg B \sqcap \neg C \sqcap \neg D \sqcap \exists H_1.B \sqcap \exists V_1.C \sqcap \leqslant 3 S_{11}),\\
\top &\doteq B \Rightarrow (\neg A \sqcap \neg C \sqcap \neg D \sqcap \exists H_2.A \sqcap \exists V_1.D \sqcap \leqslant 3 S_{21}),\\
\top &\doteq C \Rightarrow (\neg A \sqcap \neg B \sqcap \neg D \sqcap \exists H_1.D \sqcap \exists V_2.A \sqcap \leqslant 3 S_{12}),
\end{aligned}$$

…