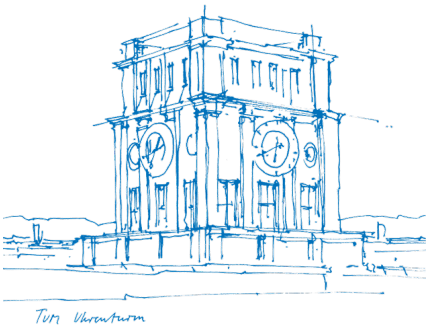
Group A
School of Computation, Information and Technology
Technical University of Munich

Hyperproperties and Temporal Logics on them

Ilya Kats, Matei Florescu, René Viehhauser

Group A
School of Computation, Information and Technology
Technical University of Munich

November 18, 2026



Outline I

1. Motivation
2. From Trace Properties to Hyperproperties
3. Hypersafety
4. Hyperliveness
5. Topology
6. Beyond Hypersafety and Hyperliveness

Outline II

7 Temporal Logics

8 Temporal Logics for Hyperproperties

Outline

1 Motivation

2 From Trace Properties to Hyperproperties

3 Hypersafety

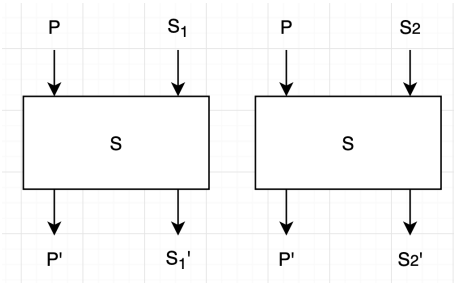
4 Hyperliveness

5 Topology

Insecurities of Hardware [6]

Spectre and *Meltdown* revealed new security vulnerabilities:

- Refinement Attacks
- Trace properties no longer suffice
- Example: *Observational Determinism*



Outline

1 Motivation

2 From Trace Properties to Hyperproperties
- Trace Properties
- Hyperproperties
- Examples

3 Hypersafety

4 Hyperliveness

5 Topology

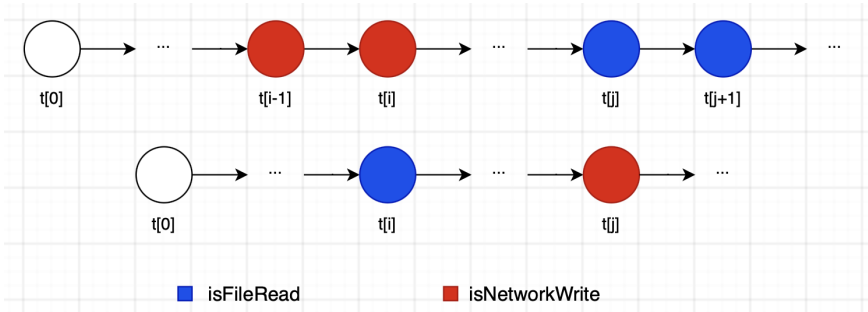
Trace properties

- Execution Traces:
  - Abstract set of states: Σ
  - Set of all finite traces: $\Psi_{fin} \triangleq \Sigma^*$
  - Set of all infinite traces: $\Psi_{inf} \triangleq \Sigma^\omega$
  - Set of all traces: $\Psi \triangleq \Psi_{fin} \cup \Psi_{inf}$
- Trace properties:
  - Prop $\triangleq \mathcal{P}(\Psi_{inf})$
  - $T \models P \triangleq T \subseteq P$

Trace properties

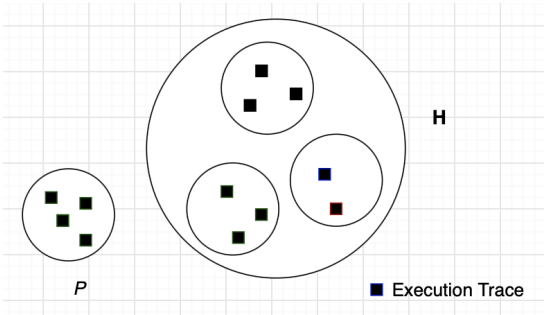
- Example:

$$\textit{NRW} \triangleq \{t \in \Psi_{inf} \mid \neg(\exists i, j \in \mathbb{N} : i < j \wedge isFileRead(t[i]) \wedge isNetworkWrite(t[j]))\}$$



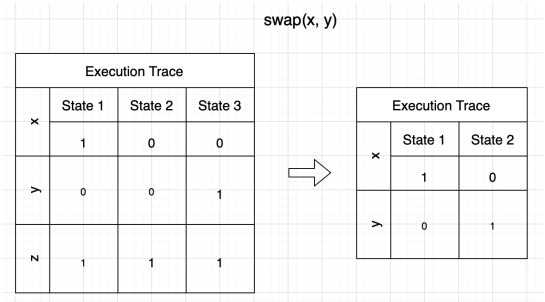
Hyperproperties [3]

- **HP** $\triangleq \mathcal{P}(\mathcal{P}(\Psi_{inf})) = \mathcal{P}(\text{Prop})$
- $T \models \boldsymbol{H} \triangleq T \in \boldsymbol{H}$



Hyperproperties

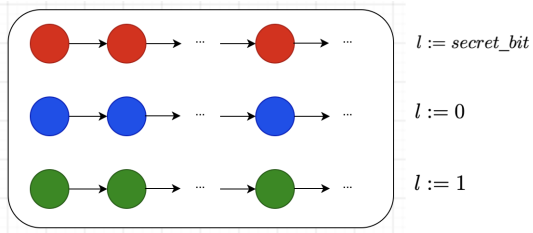
- Stepwise refinement
- Specification S_1 refines specification S_2 $\Leftrightarrow$ every behavior permitted by S_1 is also permitted by S_2
  - *Interpretation functions*: functions of type $\Psi \to \Psi$
- We can lift an interpretation function α to Prop $\to$ Prop: $\alpha(T) \triangleq \{\alpha(t) \mid t \in T\}$
- A system S α-*satisfies* trace property $P \Leftrightarrow \alpha(S) \subseteq P$



Hyperproperties

- We can lift an interpretation function α to **HP** $\rightarrow$ **HP**: $\alpha(\mathbf{H}) \triangleq \{\alpha(T) \mid T \in \mathbf{H}\}$
- H_1 α-refines $H_2 \Leftrightarrow \alpha(H_1) \subseteq H_2$
- Refinement does not generally work for security policies:

$$l := secret_bit \; [] \; l := 0 \; [] \; l := 1$$



Hyperproperties

- The set of subset-closed hyperproperties:

$$\textbf{SSC} \triangleq \{\textbf{H} \in \textbf{HP} \mid (\forall T \in \text{Prop} : T \in \textbf{H} \Rightarrow (\forall T' \in \text{Prop} : T' \subseteq T \Rightarrow T' \in \textbf{H}))\}$$

  - **SSC** is the set of hyperproperties with which refinement works
- Example of a subset-closed hyperproperty: Lift of P
  - Since they must be satisfied by the same sets of traces:

$$\begin{aligned}&(\forall T \in \text{Prop} : T \models P \Leftrightarrow T \models [P])\\ &= (\forall T \in \text{Prop} : T \subseteq P \Leftrightarrow T \in [P])\\ &= ([P] = \{T \in \text{Prop} \mid T \subseteq P\})\end{aligned}$$

  - $[P] \triangleq \mathcal{P}(P)$
  - $\forall P \in \text{Prop} : [P] \in \textbf{SSC}$

Examples

- *Generalized Noninterference*: for any traces t_1 and t_2, the system contains an interleaved t_3 with the high inputs of t_1 and low events of t_2

$$\{T \in \mathsf{Prop} \mid \forall t_1, t_2 \in T : (\exists t_3 \in T : ev_{Hin}(t_3) = ev_{Hin}(t_1) \wedge ev_L(t_3) = ev_L(t_2))\}$$

Outline

1 Motivation

2 From Trace Properties to Hyperproperties

3 Hypersafety
- Safety Properties
- Safety Hyperproperties
- k-Safety

4 Hyperliveness

5 Topology

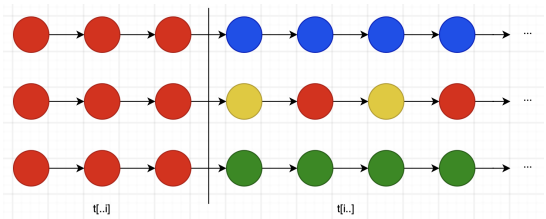
Safety Properties

- Alpern & Schneider characterize the "bad thing" in a safety property as:
  - **finitely observable**, its occurrence can be detected in finite time
  - **irremediable**, its occurrence cannot be remediated by future events
- Thus, the "bad thing" is a **finite trace** that cannot be the **prefix** of any execution in the trace property

Safety Properties

- Formally,

$$\text{S safety property} \Leftrightarrow \ (\forall t \in \Psi_{inf} : t \notin S \Rightarrow (\exists m \in \Psi_{fin} : m \preceq t \\ \wedge (\forall t' \in \Psi_{inf} : m \preceq t' \Rightarrow t' \notin S)))$$



- Define SP to be the set of all safety properties; SP is a hyperproperty

Safety Hyperproperties

- We generalize the "bad thing" from a finite trace to a finite set of finite traces, which we call **observations**
  - Obs $\triangleq \mathcal{P}^{fin}(\Psi_{fin})$
  - Prefixes on trace properties:

$$T \preceq T' := (\forall t \in T : (\exists t' \in T' : t \preceq t'))$$

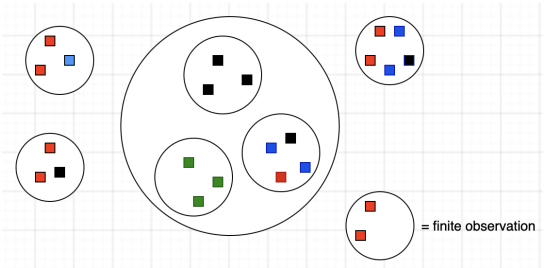
- Formal definition of a safety hyperproperty (hypersafety):

$$\begin{aligned} \textbf{S} \text{ hypersafety} \Leftrightarrow\ & (\forall T \in \text{Prop} : T \notin \textbf{S} \Rightarrow (\exists M \in \text{Obs} : M \preceq T \\ & \wedge (\forall T' \in \text{Prop} : M \preceq T' \Rightarrow T' \notin \textbf{S}))) \end{aligned}$$

- Define **SHP** to be the set of all safety hyperproperties

Safety Hyperproperties

- SP $\notin$ **SHP**: $\Psi_{inf} \in$ SP
- **SHP** $\subset$ **SSC**
  - $T \in \mathbf{H} \Rightarrow (\forall T' \subseteq T : T' \in \mathbf{H})$
  - Consequence: Stepwise refinement works on safety hyperproperties



k-safety hyperproperties

Now we introduce the definition of the k-safety hyperproperty

- A hyperproperty $\boldsymbol{S}$ is a k-safety hyperproperty (is a k-safety) iff

$$(\forall T \in \mathsf{Prop} : \; T \notin \boldsymbol{S} \implies (\exists M \in \mathsf{Obs} : M \preceq T \wedge |M| \leq k \\ \wedge (\forall T' \in \mathsf{Prop} : \; M \preceq T' \implies T' \notin \boldsymbol{S}))).$$

- The set of all k-safety hyperproperties is denoted by $\mathbf{KSHP}(k)$

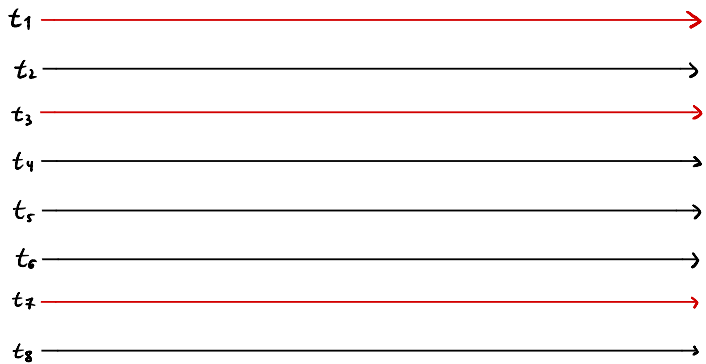
A finite witness of a violation

Here with H we denote a 3-safety hyperproperty



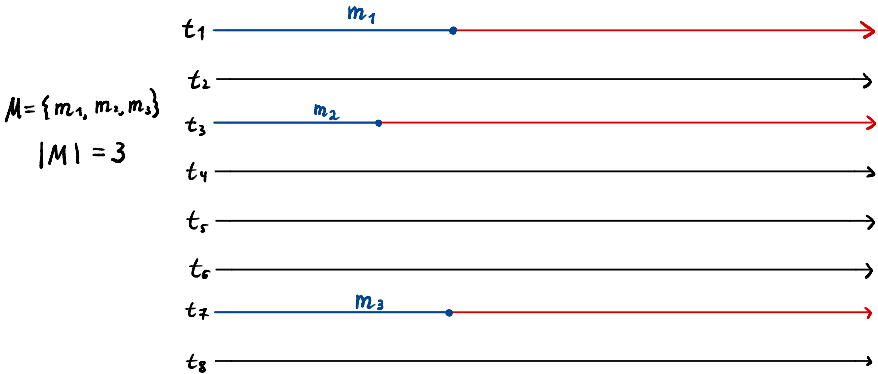
Consider execution traces of a system T.

A finite witness of a violation



Select traces whose finite prefixes witness a violation.

A finite witness of a violation



If M is a bad observation, every system extending M violates $\boldsymbol{H}$. Here, $|M| = 3$.

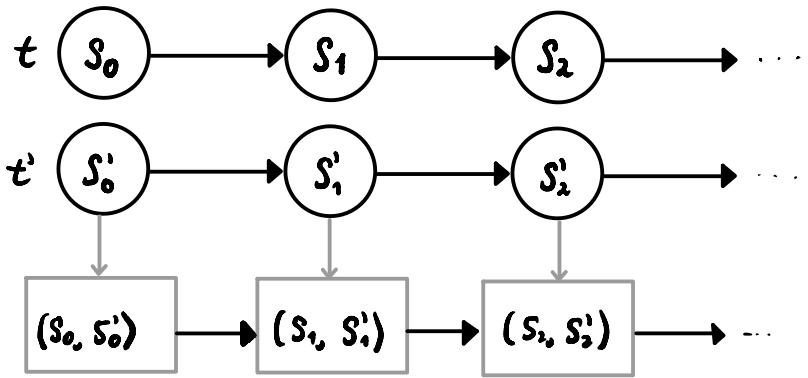
Parallel self-composition

- We define a parallel self-composition of a system S as the product system $S \times S$ consisting of traces over $\Sigma \times \Sigma$.

$$S \times S \triangleq \{(t[0], t'[0])(t[1], t'[1]) \cdots \mid t \in S \wedge t' \in S\}.$$

- The k-product, denoted S^k, is defined as a k-fold parallel self-composition of S

Parallel self-composition



Reduction to safety

- Let Sys be the set of all systems (= Prop $\setminus \{\emptyset\}$)
- Then the following holds:

$$(\forall S \in \mathsf{Sys},\ \boldsymbol{K} \in \mathbf{KSHP}(k) :\ (\exists K \in \mathsf{SP} :\ S \models \boldsymbol{K} \iff S^k \models K)).$$

Hypersafety beyond any fixed k

- $\boldsymbol{SecS}_k$ is a k-safety: a system cannot output all k shares of a secret from a k-share secret sharing scheme
- Then

$$\boldsymbol{SecS} \triangleq \bigcup_{k \geq 1} \boldsymbol{SecS}_k.$$

  is not a k-safety for any $k \in \mathbb{N}$

Outline

2 From Trace Properties to Hyperproperties

3 Hypersafety

4 Hyperliveness
- Liveness Properties
- Liveness Hyperproperties

5 Topology

6 Beyond Hypersafety and Hyperliveness

7 Temporal Logics

Liveness Properties

Liveness properties are defined by the "good thing".
Alpern and Schneider characterize the “good thing” in a liveness property as:

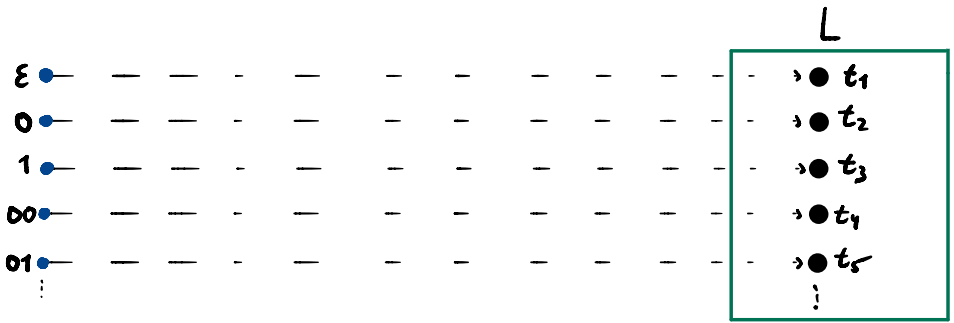
- always possible, no matter what has occurred so far, and
- possibly infinite, so it need not be a discrete event.

Formally, a trace property L is a liveness property iff

$$\left(\forall t \in \Psi_{\text{fin}} : \left(\exists t' \in \Psi_{\text{inf}} : t \preceq t' \wedge t' \in L\right)\right).$$

- We define LP as the set of all liveness properties

Liveness: every finite prefix has a "good" continuation



Every finite prefix has **at least one** infinite extension in L.

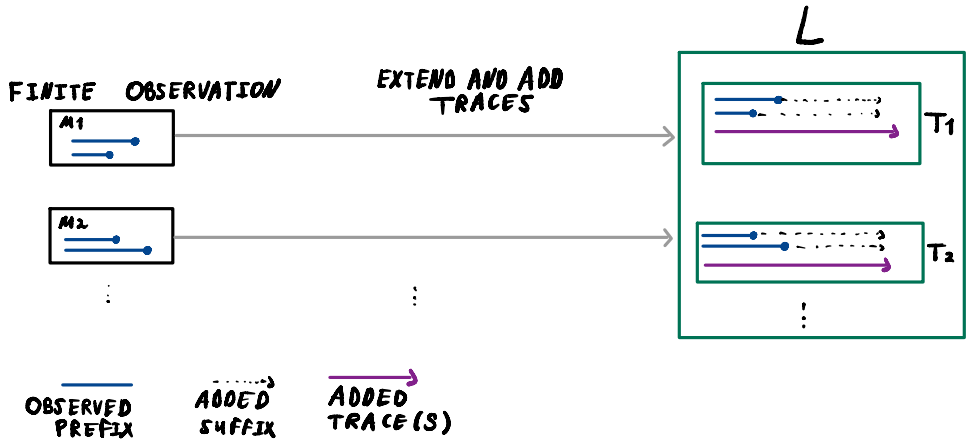
Liveness Hyperproperties

- A hyperproperty $\boldsymbol{L}$ is a liveness hyperproperty (is hyperlive) iff

$$\left(\forall T \in \mathsf{Obs} : \left(\exists T' \in \mathsf{Prop} : T \preceq T' \wedge T' \in \boldsymbol{L}\right)\right).$$

- Define **LHP** as the set of all liveness hyperproperties.

Hyperliveness



Consequences of hyperliveness

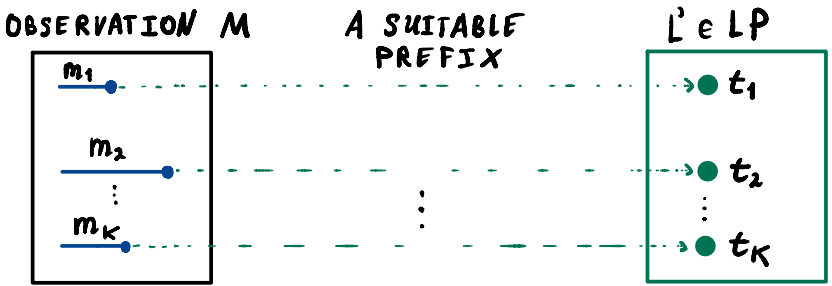
- $\textbf{LHP} \cap \textbf{SHP} = \{\mathit{true}\}$ with $\mathit{true} = \mathrm{Prop}$.
- $\boldsymbol{\mathit{false}} \in \textbf{SHP} \setminus \textbf{LHP}$ with $\boldsymbol{\mathit{false}} = \{\emptyset\}$
- Liveness properties lift to liveness hyperproperties

$$(\forall L \in \mathsf{Prop} : L \in \mathsf{LP} \iff [L] \in \textbf{LHP})\,.$$

- LP is a hyperliveness (and SP is a hyperliveness)

LP **is a hyperliveness**

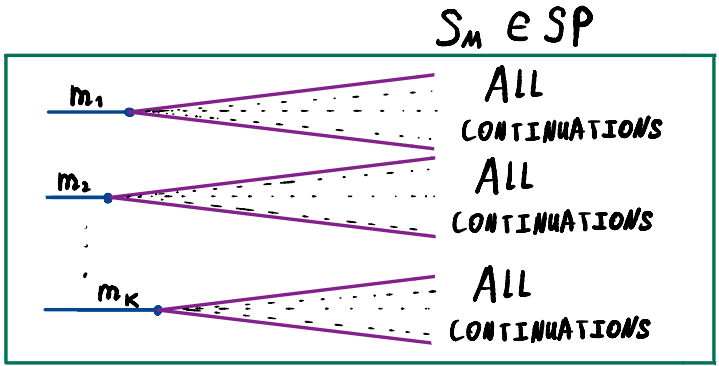
Fix any $L' \in \text{LP}$ and any finite observation M.



Every observed prefix has an infinite extension in L'. Thus $M \preceq L'$.

SP **is a hyperliveness**

Given $M = \{m_1, \ldots, m_k\}$, include **all** continuations of its prefixes.



A trace belongs to S_M exactly when it starts with one of the prefixes in M.

Outline

3 Hypersafety

4 Hyperliveness

5 Topology

6 Beyond Hypersafety and Hyperliveness

7 Temporal Logics

8 Temporal Logics for Hyperproperties

Observable properties

- An observable property is a property which can be observed in a finite amount of time
- Formally, O is an observable property iff

$$\big(\forall t \in \Psi_{\text{inf}} :\ t \in O \implies \big(\exists m \in \Psi_{\text{fin}} : m \preceq t \\ \wedge \big(\forall t' \in \Psi_{\text{inf}} :\ m \preceq t' \implies t' \in O\big)\big)\big).$$

- Let $\mathcal{O}$ be the set of observable properties. $\mathcal{O}$ satisfies two closure properties
  - If $O_1, \ldots, O_n$ are observable then their intersection is also observable
  - A union of arbitrarily many observable properties is observable

Topology

- A topology $\mathcal{T}$ on a set S defined as: $\mathcal{T} \subset \mathcal{P}(S)$ s.t. $\emptyset, S \in \mathcal{T}$;
  $\mathcal{T}$ is closed under finite intersections and arbitrary unions.
- Hence, $\mathcal{O}$ is a topology on the set of infinite traces.
  We call it Plotkin topology.
- The elements of a topology are called open sets.
- A base $B \subseteq \mathcal{T}$ satisfies

$$\forall O \in \mathcal{T}\ \exists J\ \exists (B_j)_{j \in J} \in B^J \quad O = \bigcup_{j \in J} B_j.$$

Plotkin Topology

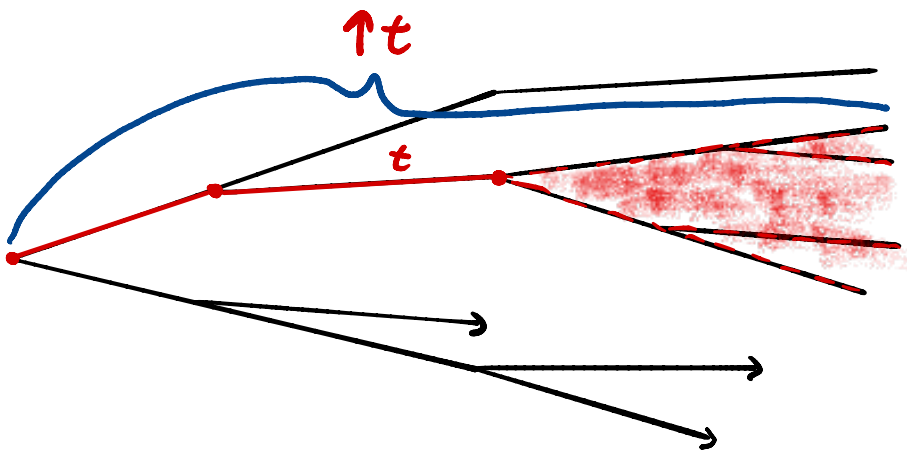
- We define

$$\uparrow t \triangleq \{t' \in \Psi_{\text{inf}} \mid t \preceq t'\}.$$

  which is the set of completions of a finite trace t
- For the topology $\mathcal{O}$:

$$\mathcal{O}^B \triangleq \{\uparrow t \mid t \in \Psi_{\text{fin}}\}$$

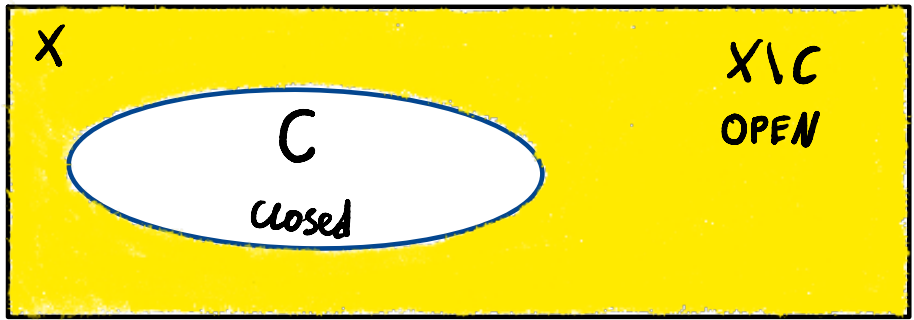
Completions of a finite trace



Safety as closed sets and liveness as dense sets

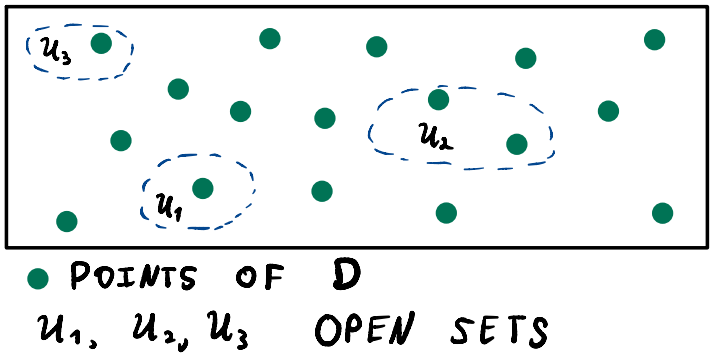
- A set C is called closed if $C^c \in \mathcal{T}$
- A set D is called dense if $\forall M \in \mathcal{T}$ s.t. $M \neq \emptyset : \ D \cap M \neq \emptyset$
- In Plotkin topology: closed sets correspond to safety properties, dense sets correspond to liveness properties

Closed sets



A set C is closed iff $X \setminus C$ is open

Dense sets



Every nonempty open set contains a point of D.

Topology on sets of traces

- Define the set of completions of an observation M as:

$$\uparrow M \triangleq \{T \in \mathsf{Prop} \mid M \preceq T\}.$$

- The topology $\mathcal{O}$ on sets of traces can be defined via its base.

$$\mathcal{O}^B \triangleq \{\uparrow M \mid M \in \mathsf{Obs}\}.$$

- Define **C** as the set of closed sets, **D** as the set of dense sets in $\mathcal{O}$.
  Then the following holds:
  - $\mathbf{SHP} = \mathbf{C}$
  - $\mathbf{LHP} = \mathbf{D}$

Outline

3 Hypersafety

4 Hyperliveness

5 Topology

6 Beyond Hypersafety and Hyperliveness

7 Temporal Logics

8 Temporal Logics for Hyperproperties

Intersections [1]

- Alpern & Schneider concluded that every trace property is the intersection of a safety property and a liveness property

$$(\forall T \in \text{Prop} : (\exists S \in \text{SP}, L \in \text{LP} : T = S \cap L))$$

- Proof. Define S the smallest SP containing T. Let $L \triangleq \neg(S \setminus T)$.
  1. L is liveness. Suppose L is not liveness and not dense; $\exists\, O$ nonempty open set : $O \subseteq \neg L$. $O \subseteq (S \setminus T) \Leftrightarrow T \subseteq (S \setminus O)$ with $(S \setminus O)$ closed set $\Leftrightarrow (S \setminus O)$ strictly smaller SP containing T, contradicting the minimality of S
  2. $S \cap L = S \cap \neg(S \setminus T) = S \cap (\neg S \cup T) = (S \cap \neg S) \cup (S \cap T) = T$

Intersections [4]

- Analogously, we can generalize this theorem, stating that every hyperproperty is the intersection of hypersafety and a hyperliveness

$$(\forall \mathbf{P} \in \mathbf{HP} : (\exists \mathbf{S} \in \mathbf{SHP}, \mathbf{L} \in \mathbf{LHP} : \mathbf{P} = \mathbf{S} \cap \mathbf{L}))$$

- Proof. Let $\mathbf{P} \in \mathbf{HP}$.
  - Safe($\mathbf{P}$) stipulates that the hyperliveness of $\mathbf{P}$ is never violated
  - Safe($\mathbf{P}$) $\triangleq \{T \in \text{Prop} \mid \forall M \in \text{Obs} : M \preceq T \Rightarrow (\exists T' \in \text{Prop} : M \preceq T' \wedge T' \in \mathbf{P})\}$
  - Live($\mathbf{P}$) stipulates that it is always possible to either satisfy $\mathbf{P}$ or become impossible
  - Live($\mathbf{P}$) $\triangleq \mathbf{P} \cup \neg$Safe($\mathbf{P}$)
  - Live($\mathbf{P}$) $\cap$ Safe($\mathbf{P}$) $= (\mathbf{P} \cup \neg\text{Safe}(\mathbf{P})) \cap \text{Safe}(\mathbf{P}) = \mathbf{P}$

Outline

4 Hyperliveness

5 Topology

6 Beyond Hypersafety and Hyperliveness

7 Temporal Logics
- LTL
- CTL and CTL*

8 Temporal Logics for Hyperproperties

Linear-time logics [2]

- any logic interpreted on sequences of valuations
- model: time progresses in discrete steps and in linear fashion, each point in time has exactly one possible future

most prominent example: LTL (Linear- Time Logic)

- in use for verification since the end of the 1970s
- specification of correctness properties

LTL Syntax

let ***AP*** be a set of atomic propositions
the set of **LTL formulas** over ***AP*** is defined inductively via

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{X}\,\varphi \mid \varphi\ \mathbf{U}\ \varphi$$

where $p \in$ ***AP***
names of the temporal operators: ne**X**t, **U**ntil

Syntactic Sugar

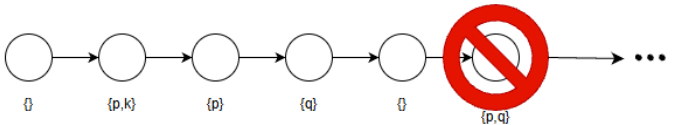
technically unneeded but simplify proofs:

$$\varphi_1 \wedge \varphi_2 \equiv \neg(\neg\varphi_1 \vee \neg\varphi_2)$$
$$\varphi_1 \rightarrow \varphi_2 \equiv \neg\varphi_1 \vee \varphi_2$$
$$tt \equiv p \vee \neg p$$
$$ff \equiv \neg tt$$
$$\mathbf{F}\,\varphi \equiv tt\ \mathbf{U}\ \varphi$$
$$\mathbf{G}\,\varphi \equiv \neg\,\mathbf{F}\,\neg\varphi$$
$$\varphi_1\ \mathbf{W}\ \varphi_2 \equiv (\varphi_1\ \mathbf{U}\ \varphi_2) \vee \mathbf{G}\,\varphi_1$$
$$\varphi_1\ \mathbf{R}\ \varphi_2 \equiv \neg(\neg\varphi_1\ \mathbf{U}\ \neg\varphi_2)$$
$$\varphi_1\ \mathbf{M}\ \varphi_2 \equiv \neg(\neg\varphi_1\ \mathbf{W}\ \neg\varphi_2)$$

names: **F**inally (eventually), **G**enerally (globally, always), **W**eak Until, **R**elease, **M**ighty Release

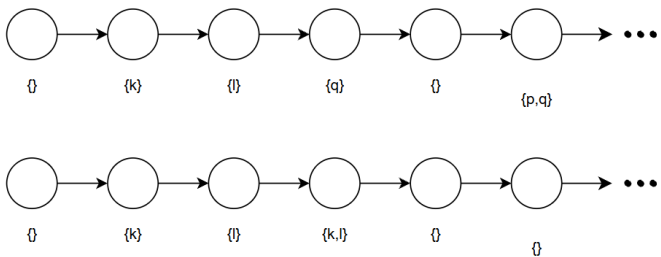
Examples

- mutex property: **G** $\neg(p \wedge q)$
  p and q are never true at the same time



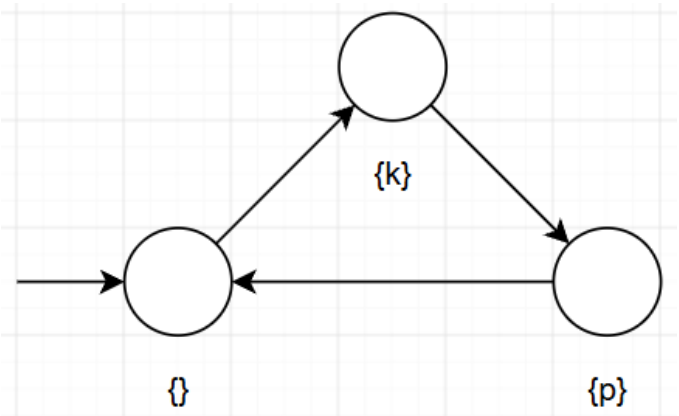
Examples

- mutex property: **G** $\neg(p \wedge q)$
  p and q are never true at the same time
- safety: $(\neg p)$ **W** q
  p does not occur before q has happened, but q might also never happen



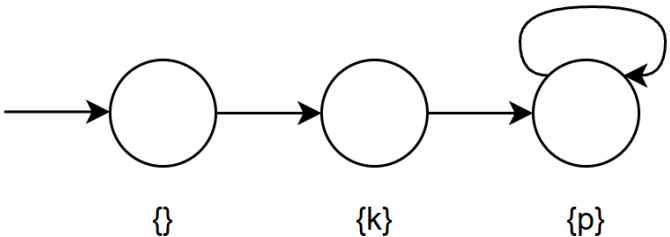
Examples

- liveness: $(\mathbf{G}\,\mathbf{F}\,p)$
  p occurs infinitely often



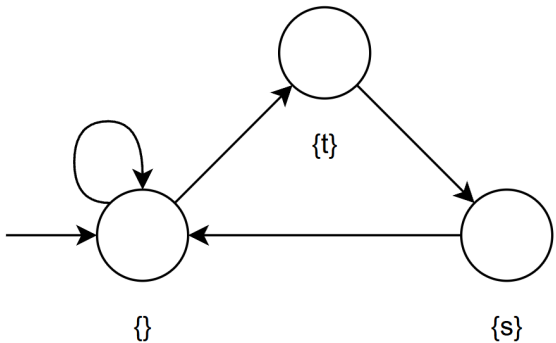
Examples

- liveness: $(\mathbf{G}\,\mathbf{F}\,p)$
  p occurs infinitely often
- $\mathbf{F}\,\mathbf{G}\,p$
  at some point, p will continue to hold forever



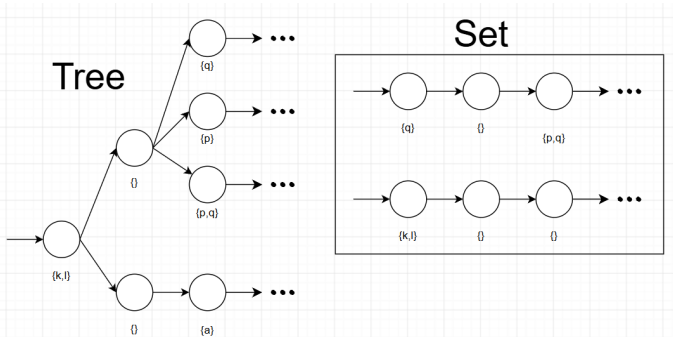
Examples

- $\mathbf{G}(t \rightarrow \mathbf{F}s)$
  whenever a process tries (t) to enter its critical section, it will eventually succeed (s)



Branching-time logics [2]

- linear-time logic describes properties of runs,
  however it cannot describe options that are possible in a system
- branching-time logics allow speaking about branching behavior, i.e. multiple possible futures
  → evaluated over trees of valuations (that's **NOT** the same thing as sets of valuations)



CTL* Syntax

extends the syntax of LTL and is defined inductively via

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi \mid \mathbf{E}\,\varphi$$

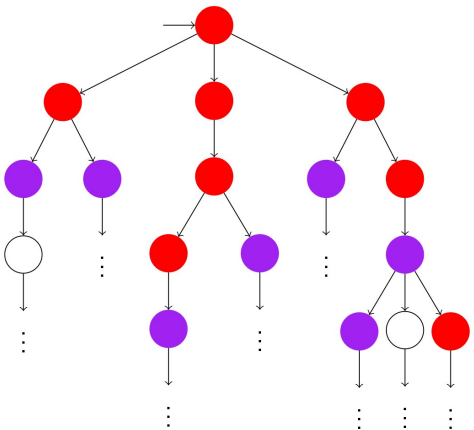
where again $p \in \boldsymbol{AP}$, a set of atomic propositions
CTL* is quite powerful but the associated model-checking problem is correspondingly hard, that power is not always needed
so CTL* is often restricted to CTL, which has the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{E}\,\mathbf{X}\,\varphi \mid \mathbf{E}(\varphi\,\mathbf{U}\,\varphi) \mid \mathbf{E}\,\mathbf{G}\,\varphi$$

Syntactic Sugar can be defined for both, similar to how we did for LTL (e.g. $\mathbf{A}\varphi \equiv \neg\mathbf{E}\neg\varphi$)

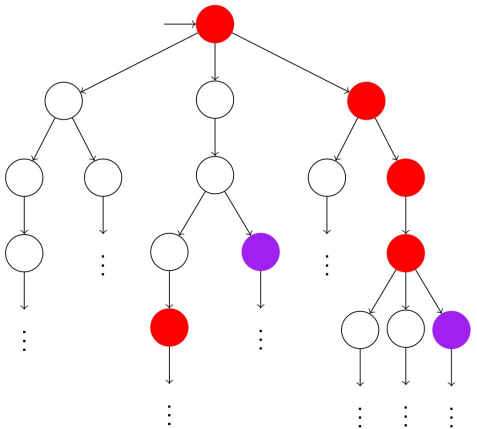
Examples

formula: $\mathtt{A}(\mathtt{red}\ \mathtt{U}\ \mathtt{purple})$



Examples

formula: $\mathtt{E}(\mathtt{red}\ \mathtt{U}\ \mathtt{purple})$



Relations between the three Logics

- CTL* is by definition a superset of LTL (Remark: When taking LTL as a subset of CTL*, any LTL formula is implicitly prefixed with the universal path quantifier **A** [All path Semantic])
- CTL is by definition a subset of CTL*
- CTL and LTL are incomparable, as LTL can say things CTL can't and vice versa
- Examples:
  - formula that is in both: **A G** p
  - LTL formula that is not in CTL: **A F G** p
  - CTL formula that is not in LTL: **E X** p
  - CTL* formula that is in neither: **E X** $p \wedge$ **A F G** p

Why CTL* is not enough for Hyperproperties [5]

- Even though CTL* has explicit path quantifiers, information-flow security policies cannot be expressed with them
- consider:

$s \models \mathbf{A}\ \mathbf{A}\ \varphi$
= for all $p \in$ Paths(K,s) : $p \models \mathbf{A}\ \varphi$
= for all $p \in$ Paths(K,s) and $p' \in$ Paths(K,s) : $p' \models \varphi$

- The meaning of the formula is determined by φ, which is modeled by the single path p'
- Path p is ignored in determining the meaning as the second universal quantifier causes p to "leave scope"
- Hence φ cannot express correlations between p and p'

Outline

4 Hyperliveness

5 Topology

6 Beyond Hypersafety and Hyperliveness

7 Temporal Logics

8 Temporal Logics for Hyperproperties
- HyperLTL
- HyperCTL*

Hyper LTL [5]

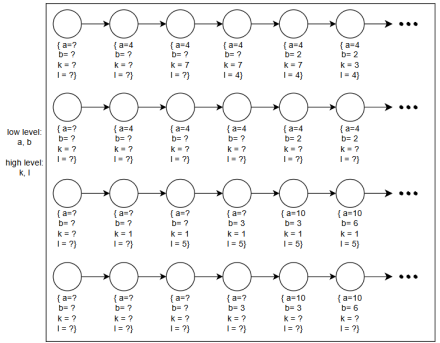
- HyperLTL extends LTL with explicit quantification over traces
- Let π be a *trace variable* from an infinite supply υ of trace variables and a an atomic proposition, then the syntax of HyperLTL is defined via:

$$\psi ::= \exists \pi.\psi \mid \forall \pi.\psi \mid \varphi$$
$$\varphi ::= a_\pi \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{X}\, \varphi \mid \varphi\, \mathbf{U}\, \varphi$$

- As an example: $\forall \pi_1.\ \forall \pi_2.\ \exists \pi_3.\psi$
  means that for all traces π_1 and π_2 , there exists another trace π_3 , such that ψ holds on those three traces
- Another Example, given a set *P* of atomic propositions: **G** $(\pi[0] =_P \pi'[0])$
  means that all the positions of π and π' agree on *P*

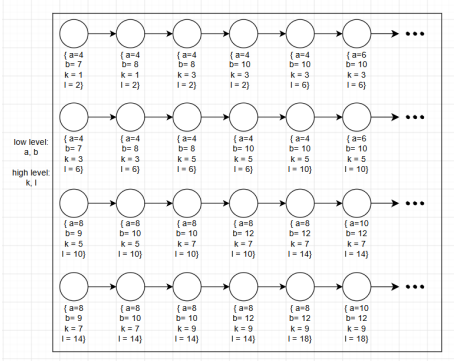
Examples of Security Policies in HyperLTL

- Noninference (Liveness Hyperproperty): $\forall\pi.\exists\pi'.(\mathbf{G}\ \lambda_{\pi'})\ \wedge \pi =_L \pi'$
  for all traces, the low-observable behaviour must not change ($=_L$) when all high inputs are replaced with dummy input λ ($\lambda_{\pi'}$), that is, when the high input is removed



Examples of Security Policies in HyperLTL

- Observational determinism (Safety Hyperproperty):
  $\forall \pi . \forall \pi' . \pi[0] =_{L,in} \pi'[0] \rightarrow \pi =_{L,out} \pi'$
  every pair of traces with the same initial low observation remain indistinguishable



HyperCTL* [5]

- HyperLTL was derived from LTL by extending it from single traces to sets of traces
- Because of that also restricted to linear time and cannot express branching-time properties
- Therefore derive a new logic for Hyperproperties from CTL*
- Let π be a *path variable* from an infinite supply υ of path variables and a an atomic proposition, then the syntax of HyperCTL* is defined via:

$$\varphi ::= a_\pi \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{X}\, \varphi \mid \varphi\, \mathbf{U}\, \varphi \mid \exists \pi.\varphi$$

Relationship between HyperCTL* vs. HyperLTL

- By definition HyperLTL is a subset of HyperCTL*
- In particular HyperLTL can be characterized as the fragment of HyperCTL* containing formulas in *prenex* form, that is, a series of quantifiers followed by a quantifier-free formula
- Just as CTL* is a true superset of LTL, HyperCTL* is a true superset of HyperLTL
- As an example the following formula: $\forall \pi.\ \mathbf{X}\ \forall \pi'.\ \mathbf{X}\ (l_\pi \leftrightarrow l_{\pi'})$
  cannot be expressed in HyperLTL

Outline I

1. Motivation
2. From Trace Properties to Hyperproperties
3. Hypersafety
4. Hyperliveness
5. Topology
6. Beyond Hypersafety and Hyperliveness

Outline II

7 Temporal Logics

8 Temporal Logics for Hyperproperties
- HyperLTL
- HyperCTL*

Sources

[1] Bowen Alpern and Fred B. Schneider. *Defining Liveness | Information Processing Letters, Volume 21, Issue 4, 1985, Pages 181-185.* Feb. 1985. URL: https://www.cs.cornell.edu/fbs/publications/DefLiveness.pdf.

[2] Florian Bruse. *Model Checking VL Summerterm 2026*. 2026.

[3] Michael R. Clarkson. *Hyperproperties Course Springterm 2016*. 2016.

[4] Michael R. Clarkson and Fred B. Schneider. *Hyperproperties | Journal of Computer Security*. Sept. 2010. URL: https://dl.acm.org/doi/abs/10.5555/1891823.1891830.

[5] Michael R. Clarkson et al. *Temporal Logics for Hyperproperties*. 2014. arXiv: 1401.4492 [cs.LO]. URL: https://arxiv.org/abs/1401.4492.

[6] Andrew Myers. *Meltdown, Spectre, and why hardware can be correct yet insecure*. 2018. URL: https://andrumyers.wordpress.com/2018/01/17/meltdown-spectre-and-how-hardware-can-be-correct-but-insecure/.