CS 5430

Hyperproperties

Prof. Clarkson
Spring 2016

Security Policies Today

Confidentiality

"Protection of assets from unauthorized disclosure"

Integrity

"Protection of assets from unauthorized modification"

Availability

"Protection of assets from loss of use"

Formalize and verify any security policy? ✗

Program Correctness ca. 1970s

- Partial correctness (If program terminates, it produces correct output)
- Termination
- Total correctness (Program terminates and produces correct output)
- Mutual exclusion
- Deadlock freedom
- Starvation freedom

???

Safety and Liveness Properties

Intuition [Lamport 1977]:

Safety:
"Nothing bad happens"

- Partial correctness
  Bad thing: program terminates with incorrect output
- Access control
  Bad thing: subject completes operation without required rights

Liveness:
"Something good happens"

- Termination
  Good thing: termination
- Guaranteed service
  Good thing: service rendered

Properties

Trace: Sequence of execution states

$t = s_0 s_1 \ldots$

Property: Set of infinite traces

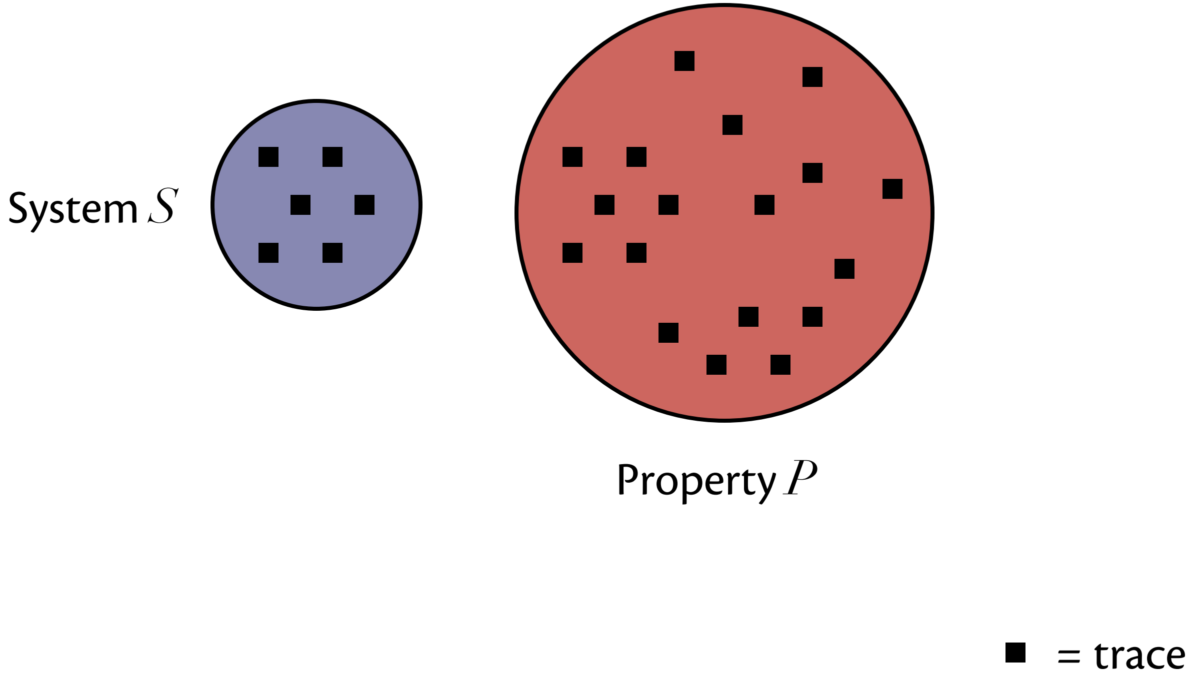
Trace t satisfies property P iff t is an element of P

➔Satisfaction depends on the trace alone

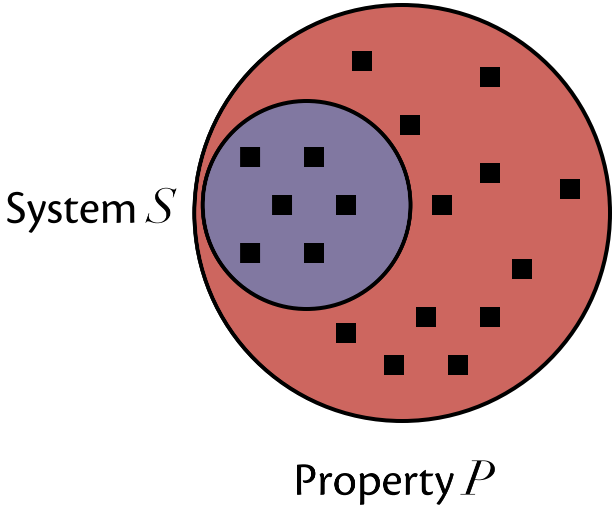
System: Also a set of traces

System S satisfies property P iff all traces of S satisfy P

Properties



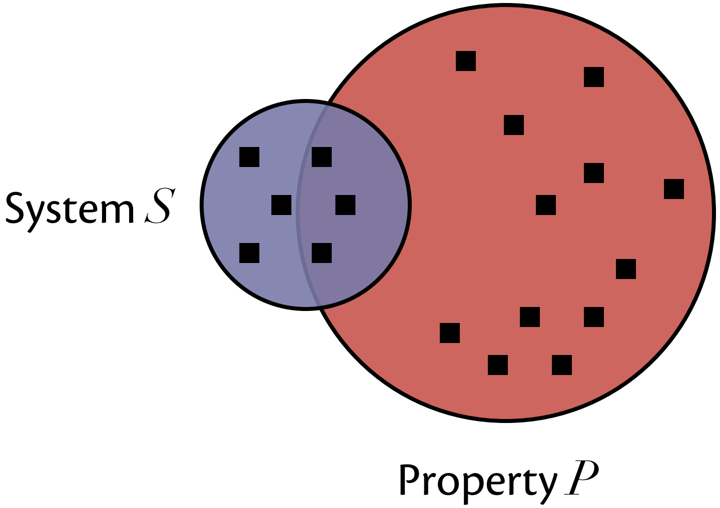
Properties



S satisfies P

■ = trace

Properties



S does not satisfy P

■ = trace

Safety and Liveness Properties

Formalized:

Safety property [Lamport 1985]

Bad thing = trace prefix

Liveness property [Alpern and Schneider 1985]

Good thing = trace suffix

Success!

Alpern and Schneider (1985, 1987):

Theorem. *Every property is the intersection of a safety property and a liveness property.*

Theorem. *Safety proved by invariance.*

Theorem. *Liveness proved by well-foundedness.*

Theorem. *Topological characterization:*

Safety = closed sets

Liveness = dense sets

Formalize and verify any property? ✓

Back to Security Policies

Formalize and verify any property? ✓

Formalize and verify any security policy? ✗

Security policy $\stackrel{?}{=}$ Property

Information Flow is not a Property

Secure information flow:

Secret inputs are not leaked to public outputs

```
p := 1;
```
✓

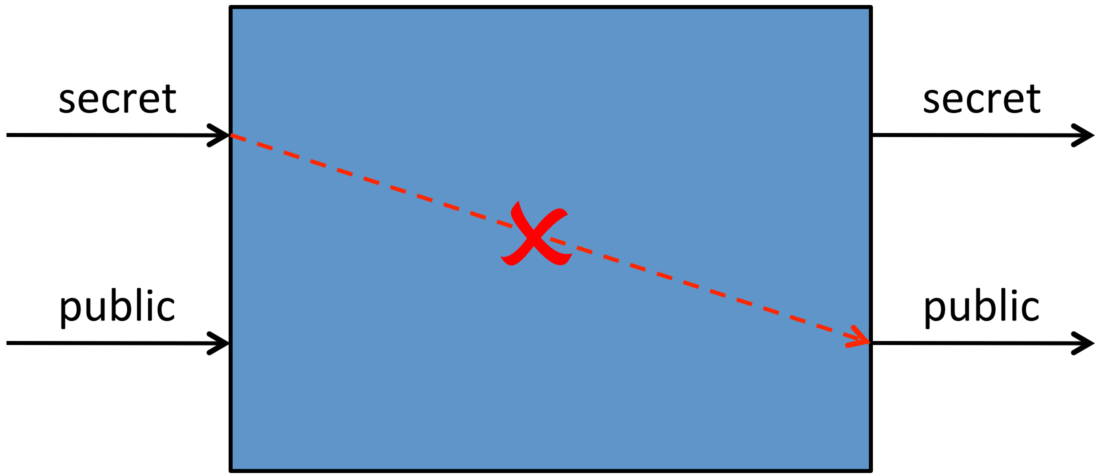
```
p := s;
```
✗

```
if (s) then p := 1 else p := 0;
```
✗

```
if (s) then {consume power} else {don't};
```
✗

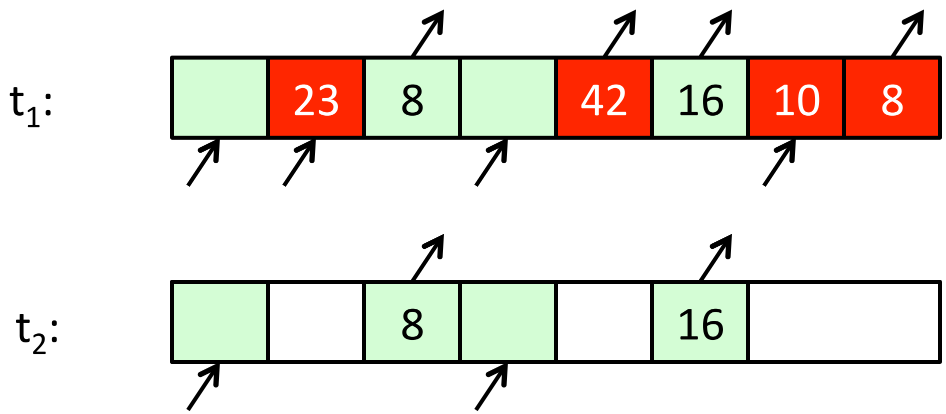
Information Flow is not a Property

Secure information flow:
Secret inputs are not leaked to public outputs



Information Flow is not a Property

Noninterference [Goguen and Meseguer 1982]: Commands of high security users have no effect on observations of low security users



Not safety!

Satisfaction depends on pairs of traces ...so not a property

Service Level Agreements are not Properties

Service level agreement: Acceptable performance of system

Not liveness!

Average response time: Average time, over all executions, to respond to request has given bound

- Satisfaction depends on all traces of system ...not a property

Any security policy that stipulates relations among traces is not a property

➔ Need satisfaction to depend on *sets* of traces [McLean 1996]

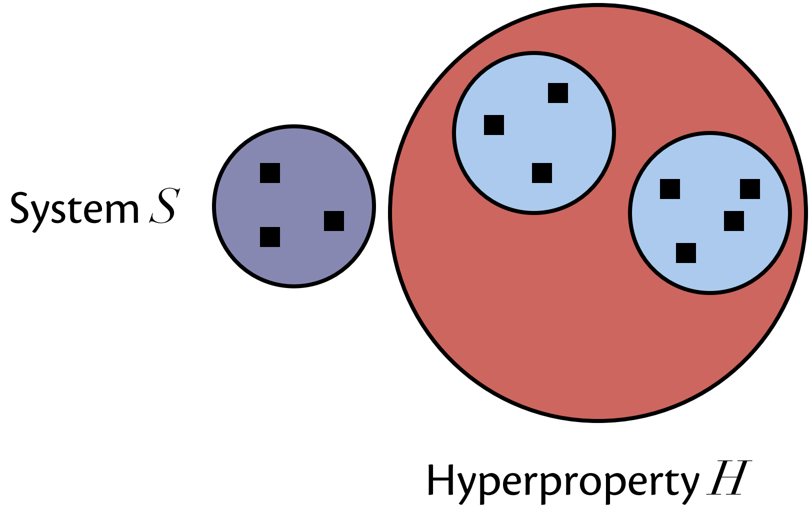
Hyperproperties

A **hyperproperty** is a set of properties
[Clarkson and Schneider 2008, 2010]

A system S **satisfies** a hyperproperty H
iff S is an element of H

…a hyperproperty specifies exactly the allowed sets of traces

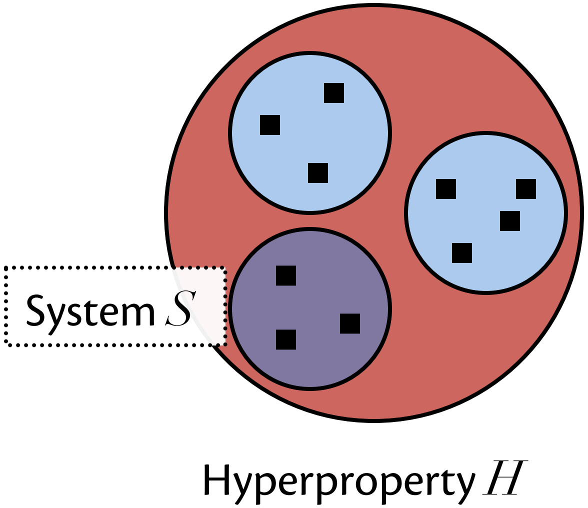
Hyperproperties



S does not satisfy H

■ = trace

Hyperproperties



S satisfies H

■ = trace

Hyperproperties

Security policies are hyperproperties!

- **Information flow:** Noninterference, relational noninterference, generalized noninterference, observational determinism, self-bisimilarity, probabilistic noninterference, quantitative leakage
- **Service-level agreements:** Average response time, time service factor, percentage uptime
- …

Beyond Hyperproperties?

- *Security policies* are predicates on *systems*
- Hyperproperties are the *extensions* of those predicates

➔Hyperproperties are expressively complete

(for predicates, systems, and trace semantics)

Other System Models

- Relational semantics
- Labeled transition systems
- State machines
- Probabilistic systems

…can define hyperproperties for all these

Probabilistic Hyperproperties

To incorporate probability:

- Assume probability on state transitions
- Construct probability measure on traces [Halpern 2003]
- Use measure to express hyperproperties

We've expressed:

- **Probabilistic noninterference** [Gray and Syverson 1998]
- Quantitative leakage
- Channel capacity

Hyperproperties

- Safety and liveness?
- Verification?

Safety

Safety proscribes "bad things"

- A bad thing is finitely observable and irremediable
- S is a safety property [Lamport 85] iff

$$(\forall\, t \notin S : (\exists\, b \leq t : (\forall\, u \geq b : u \notin S)))$$

b is a finite trace

$\notin S$

Safety

Safety proscribes "bad things"

– A bad thing is finitely observable and irremediable

– S is a safety property [Lamport 85] iff

$$(\forall\, t \notin S : (\exists\, b \leq t : (\forall\, u \geq b : u \notin S)))$$

b is a finite trace

Safety

Safety proscribes "bad things"

- A bad thing is finitely observable and irremediable
- S is a safety property [Lamport 85] iff

$$(\forall\, t \notin S : (\exists\, b \leq t : (\forall\, u \geq b : u \notin S)))$$

b is a finite trace

- S is a **safety hyperproperty** ("hypersafety") iff

$$(\forall\, T \notin \boldsymbol{S} : (\exists\, B \leq T : (\forall\, U \geq B : U \notin \boldsymbol{S})))$$

B is a finite set of finite traces

Prefix Ordering

An **observation** is a finite set of finite traces

Intuition: Observer sees a set of partial executions

$M \leq T$ (M is a **prefix** of T) iff:

- M is an observation, and
- $\forall m \in M : (\exists t \in T : m \leq t)$
- If observer watched longer, M could become T

Safety Hyperproperties

Noninterference [Goguen and Meseguer 1982]

Bad thing is a pair of traces where removing high commands does change low observations

Observational determinism [Roscoe 1995, Zdancewic and Myers 2003]

Bad thing is a pair of traces that cause system to look nondeterministic to low observer

…

Liveness

Liveness prescribes "good things"

- A good thing is always possible and possibly infinite
- L is a liveness property [AS85] iff

$$(\forall\, t : (\exists\, g \geq t : g \in L))$$

t is a finite trace

Liveness

Liveness prescribes "good things"

- A good thing is always possible and possibly infinite
- L is a liveness property [AS85] iff

$$(\forall t : (\exists g \geq t : g \in L))$$

t is a finite trace

- L is a **liveness hyperproperty** ("hyperliveness") iff

$$(\forall T : (\exists G \geq T : G \in \boldsymbol{L}))$$

T is a finite set of finite traces

Liveness Hyperproperties

Average response time

Good thing is that average time is low enough

Possibilistic information flow

Class of policies requiring "alternate possible explanations" to exist

e.g. **noninference**

Theorem. *All PIF policies are hyperliveness.*

Relating Properties and Hyperproperties

Can **lift** property T to hyperproperty $[T]$

Satisfaction is equivalent iff $[T]$ = powerset(T)

Theorem. S *is safety implies* $[S]$ *is hypersafety.*

Theorem. L *is liveness implies* $[L]$ *is hyperliveness.*

…Verification techniques for safety and liveness carry forward to hyperproperties

Safety and Liveness is a Basis (still)

Theorem. *Every hyperproperty is the intersection of a safety hyperproperty and a liveness hyperproperty.*

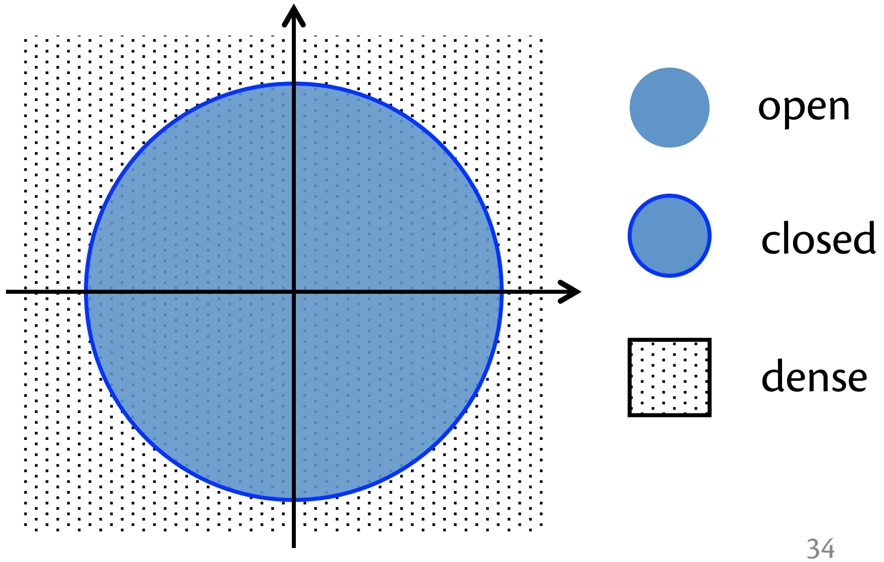
A fundamental basis…

Topology

Open set: Can always "wiggle" from point and stay in set

Closed set: "Wiggle" might move outside set

Dense set: Can always "wiggle" to get into set



Topology of Hyperproperties

For **Plotkin topology** on properties [AS85]:

- Safety = closed sets
- Liveness = dense sets

Theorem. *Hypersafety = closed sets.*

Theorem. *Hyperliveness = dense sets.*

Theorem. *Our topology on hyperproperties is equivalent to the* lower Vietoris construction *applied to the Plotkin topology.*

Stepping Back…

- Safety and liveness? ✓
- Verification?

Logic and Verification

Temporal logic: LTL, CTL*?

- Highly successful for trace properties
- But not for security policies [McLean 1994, Alur et al. 2006]
- Let's hyper-ize… with quantification over multiple traces

Syntax

LTL: [Pnueli 1977]

$\phi ::= \; p \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \ldots \mid X\,\phi \mid \phi_1\, U\, \phi_2 \mid \ldots \mid G\,\phi \mid \ldots$

State propositions: x-equals-42

HyperLTL: [Koleini, Clarkson, Micinski 2013]

$\psi ::= \; A t\!: \psi \mid E t\!: \psi \mid \phi$

State propositions annotated with trace variable: x-equals-42_t

…LTL is a fragment of HyperLTL

Examples

Observational determinism [Zdancewic and Myers 2003]:

$$\mathsf{A}t{:}\ \mathsf{A}u{:}\ t[0] =_L u[0] \Rightarrow t =_L u$$

$t[0] =_L u[0]$ is sugar for $\bigwedge_{p \in L} p_t \Leftrightarrow p_u$
(first state in both traces agrees on all propositions in L)

$t =_L u$ is sugar for $G\ (t[0] =_L u[0])$
(both traces agrees on all propositions in L)

Note: multiple paths in scope; syntax that reads like the "normal" math written in noninteference papers.

Examples

Noninference [McLean 1994]:

$$\forall t: \exists u: t =_L u \land G\ \text{no-high}_u$$

state-based variant of GM noninterference

Can also express noninterference itself.

And GNI, restrictiveness, separability, forward correctability…

Semantics

LTL:

- formula modeled by single trace: $t \vDash \phi$
- system modeled by set T of traces

HyperLTL:

- formula modeled by **set** of traces *(actually, set of named traces i.e. valuation or environment)*
- system still modeled by set T of traces, which is what quantifiers range over:

$\Pi \vDash \mathrm{A}t : \psi$ iff for all τ in T, have $\Pi, t{=}\tau \vDash \psi$

Semantics

$\Pi \vDash At : \psi$ iff for all τ in T, have $\Pi, t=\tau \vDash \psi$

$\Pi \vDash Et : \psi$ iff exists τ in T, s.t. $\Pi, t=\tau \vDash \psi$

$\Pi \vDash p_t$ iff $p \in \Pi(t)[0]$

$\Pi \vDash \neg\phi$ iff $\Pi \vDash \phi$ doesn't hold

$\Pi \vDash \phi_1 \vee \phi_2$ iff $\Pi \vDash \phi_1$ or $\Pi \vDash \phi_2$

$\Pi \vDash X\, \phi$ iff $\Pi[1..] \vDash \phi$

$\Pi \vDash \phi_1\, U\, \phi_2$ iff there exists $i \geq 0$ s.t. $\Pi[i..] \vDash \phi_2$ and for all j where $0 \leq j < i$, have $\Pi[j..] \vDash \phi_1$

Model Checking

- Adapts LTL algorithm based on Büchi automata
  [Wolper et al. 1983, Lichtenstein and Pnueli 1985, Vardi and Wolper 1994, ...]
- Prototype...
  - builds automata using self-composition [Barthe et al. 2004],
  - then outsources to GOAL [Tsay et al. 2007] for automata constructions
- Supports fragment of HyperLTL
  - Up to one quantifier alternation, e.g. AE, AAE, EA
  - Suffices for all our information-flow examples
- Yields verification methodology for any *linear-time* hyperproperty

Model Checking: Complexity

- Fragment with 1 alternation:
  - Exponential in size of system and
  - Doubly exponential in size of formula

- Full HyperLTL:
  - PSPACE-hard
  - Reduction from quantified propositional temporal logic (QPTL)

…price of security? Or do we need to be more clever?

Other Hyper Temporal Logics

- **HyperCTL*** [Finkbeiner et al. 2013]
  - Like HyperLTL, but quantifiers can be nested
  - Model checking is NSPACE(f(size of system))-complete

    where f involves a tower of exponentials… ☹

- **“Hyper modal μ-calculus”**
  - Polyadic modal μ-calculus [Andersen 1994]
  - Used by Milushev and Clarke [2012] for *incremental hyperproperties*

Stepping Back…

- Safety and liveness? ✓
- Verification?
  - Model-checking (expensive) ✓
  - Reduce to trace properties
  - Refinement

Verification of 2-Safety

2-safety: "Property that can be refuted by observing two finite traces" [Terauchi and Aiken 2005]

Methodology:

- Transform system with **self-composition construction** [Barthe, D'Argenio, and Rezk 2004]
- Verify safety property of transformed system
    - Implies 2-safety property of original system

…Reduction from hyperproperty to property

k-Safety Hyperproperties

A ***k*-safety hyperproperty** is a safety hyperproperty in which the bad thing never has more than *k* traces

$$(\forall T \notin \boldsymbol{S} : (\exists B \leq T : |B| \leq k \;\wedge\; (\forall U \geq B : B \notin \boldsymbol{S})))$$

Examples:

- **1-hypersafety:** the lifted safety properties
- **2-hypersafety:** Terauchi and Aiken's 2-safety properties
- ***k*-hypersafety:** $SEC(k)$ = "System can't, across all runs, output all shares of a *k*-secret sharing"
- **Not *k*-hypersafety for any *k*:** $SEC = \bigcup_k SEC(k)$

Verifying *k*-Hypersafety

Theorem. *Any k-safety hyperproperty of* S *is equivalent to a safety property of* S^k.

➔ Yields methodology for *k*-hypersafety

- Incomplete for hypersafety
- Hyperliveness? In general?

Refinement Revisited

Stepwise refinement:

- Development methodology for properties
  - Start with specification and high-level (abstract) program
  - Repeatedly **refine** program to lower-level (concrete) program
- Techniques for refinement well-developed

Long-known those techniques don't work for security policies—i.e., hyperproperties

- Develop new techniques?
- Reuse known techniques?

Refinement Revisited

Theorem. *Known techniques work with all hyperproperties that are* subset-closed.

Theorem. *All safety hyperproperties are subset-closed.*

➔Stepwise refinement applicable with hypersafety

Hyperliveness? In general?

Stepping Back…

- Safety and liveness? ✓
- Verification?
  - Model-checking (expensive) ✓
  - Reduce to trace properties (*k*-safety) ✓
  - Refinement (hypersafety) ✓
  - Proof system? (ongoing work with Hunter Goldstein)

…verify by decomposing to safety+liveness?

Summary

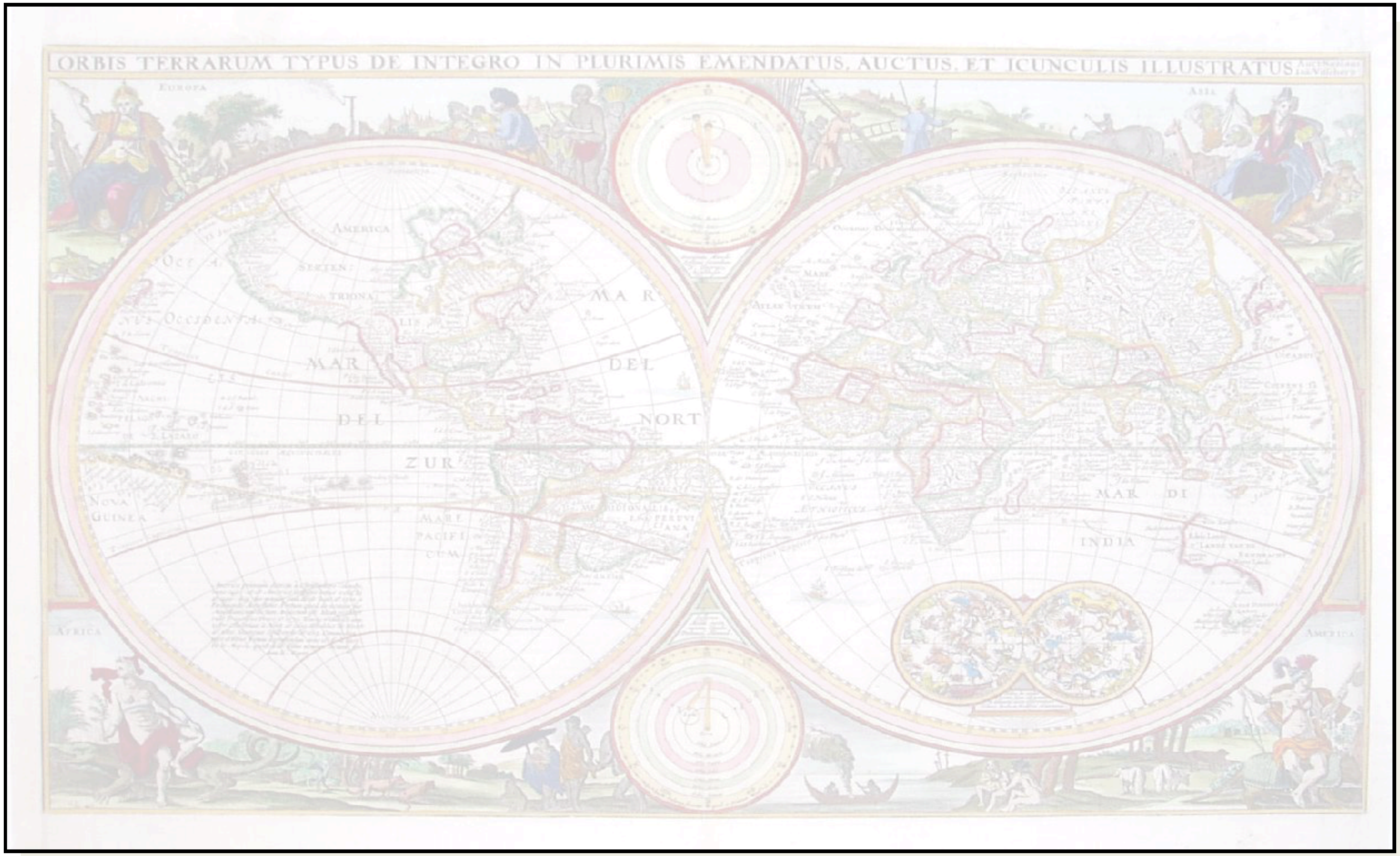
Theory of hyperproperties :

- Parallels theory of properties
  - Safety, liveness (basis, topological characterization)
  - Verification (HyperLTL, *k*-hypersafety, stepwise refinement)
- Expressive completeness
- Enables classification of security policies…

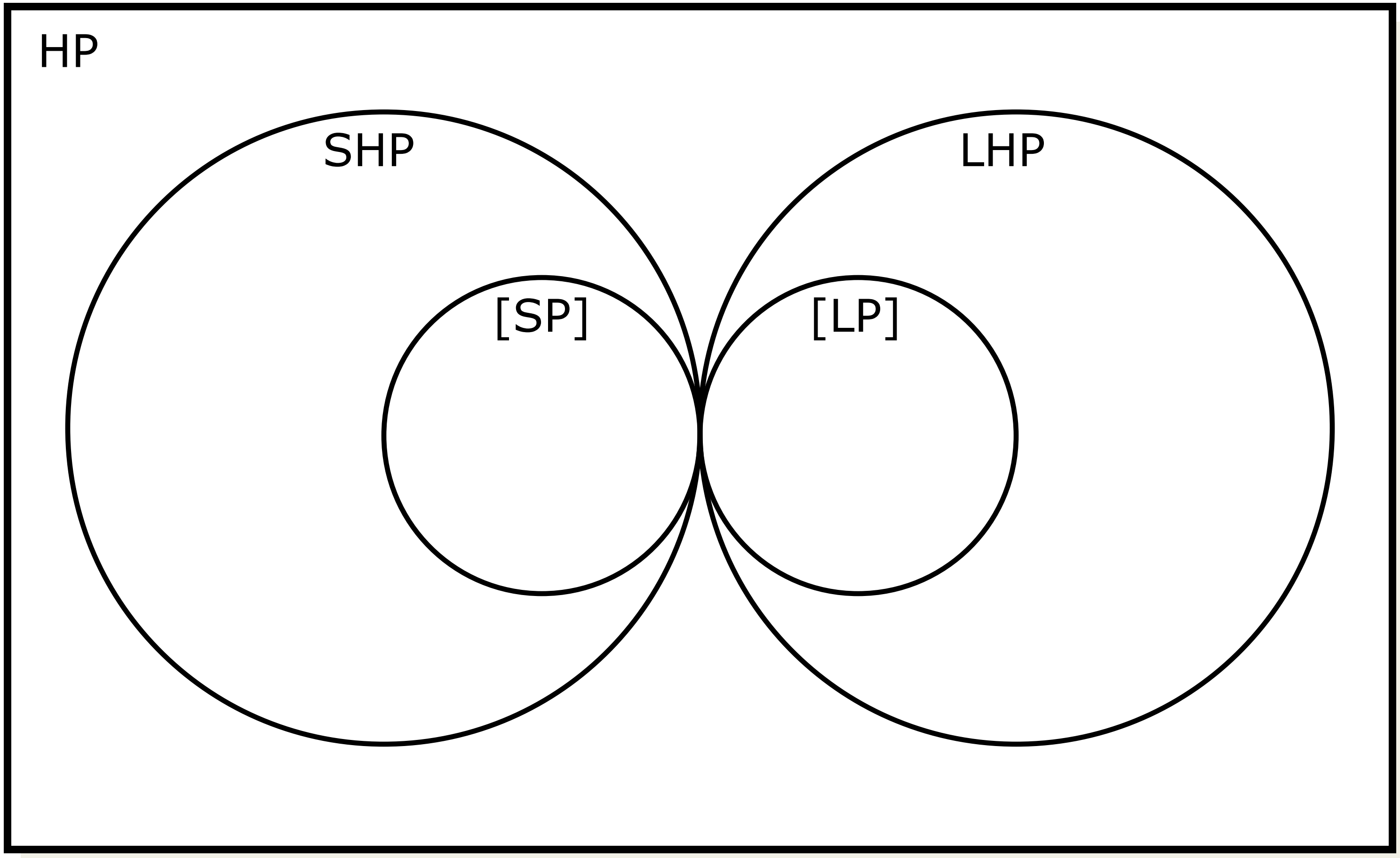
Charting the landscape…

HP

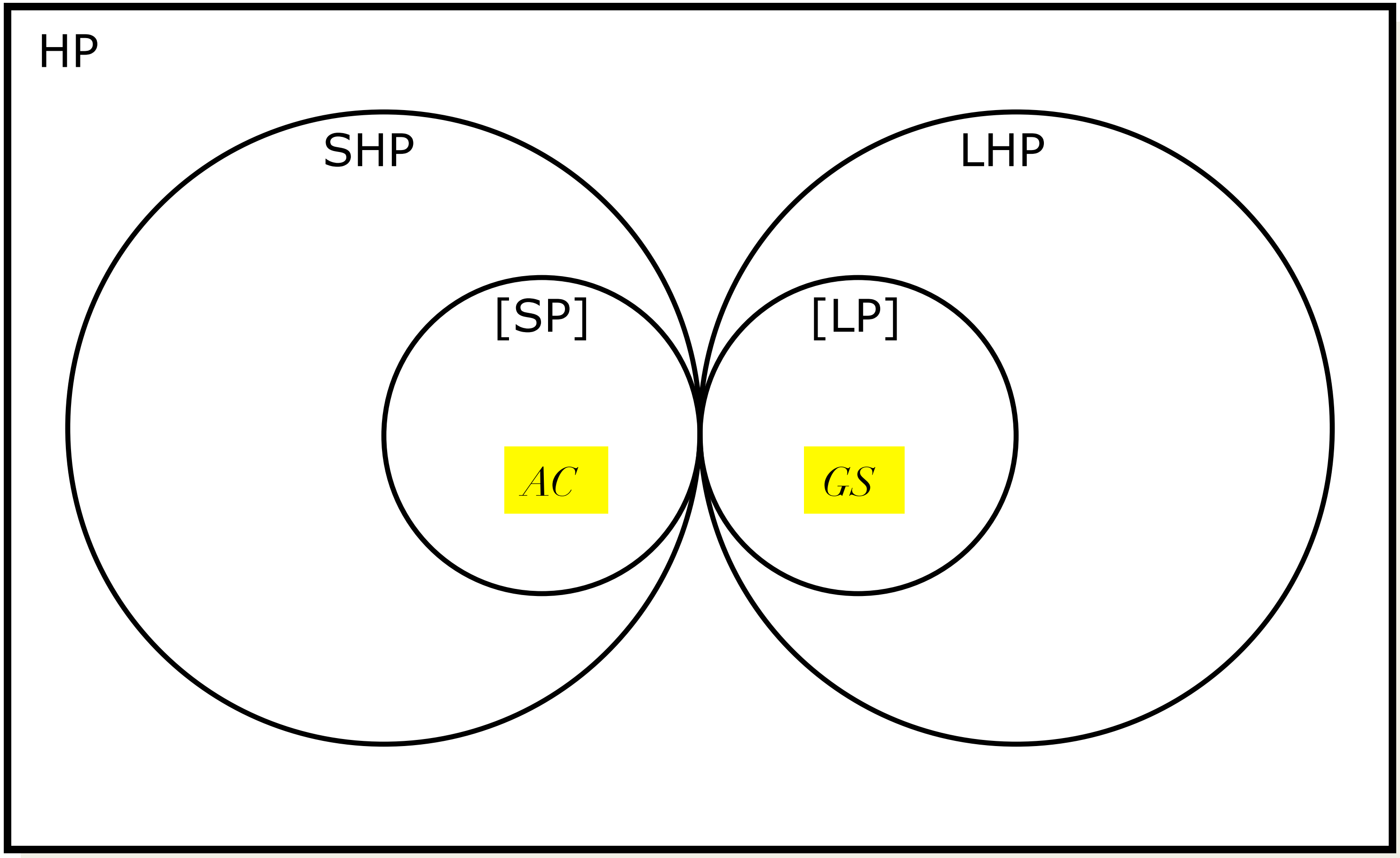
All hyperproperties (HP)



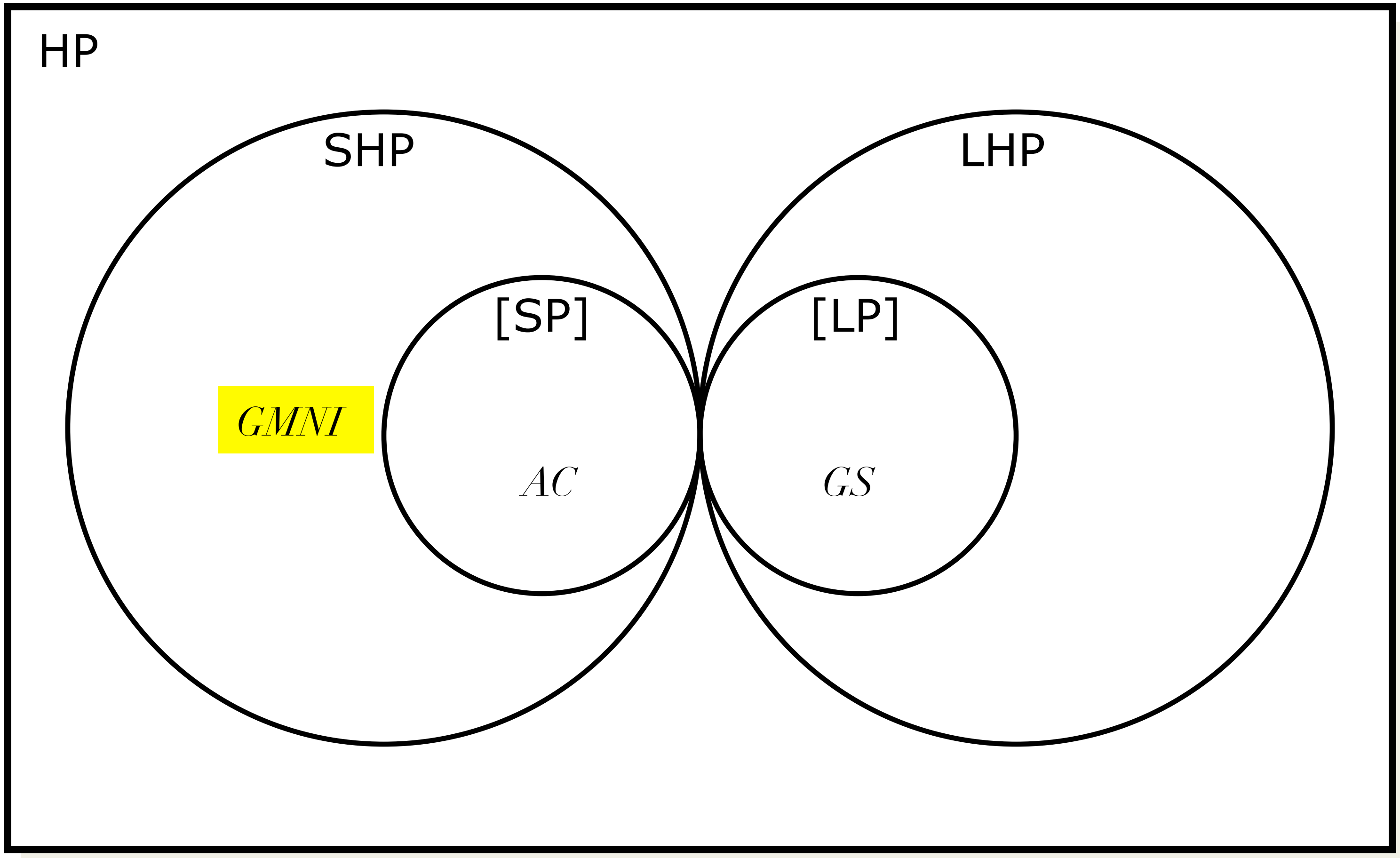
Safety hyperproperties (SHP)
Liveness hyperproperties (LHP)



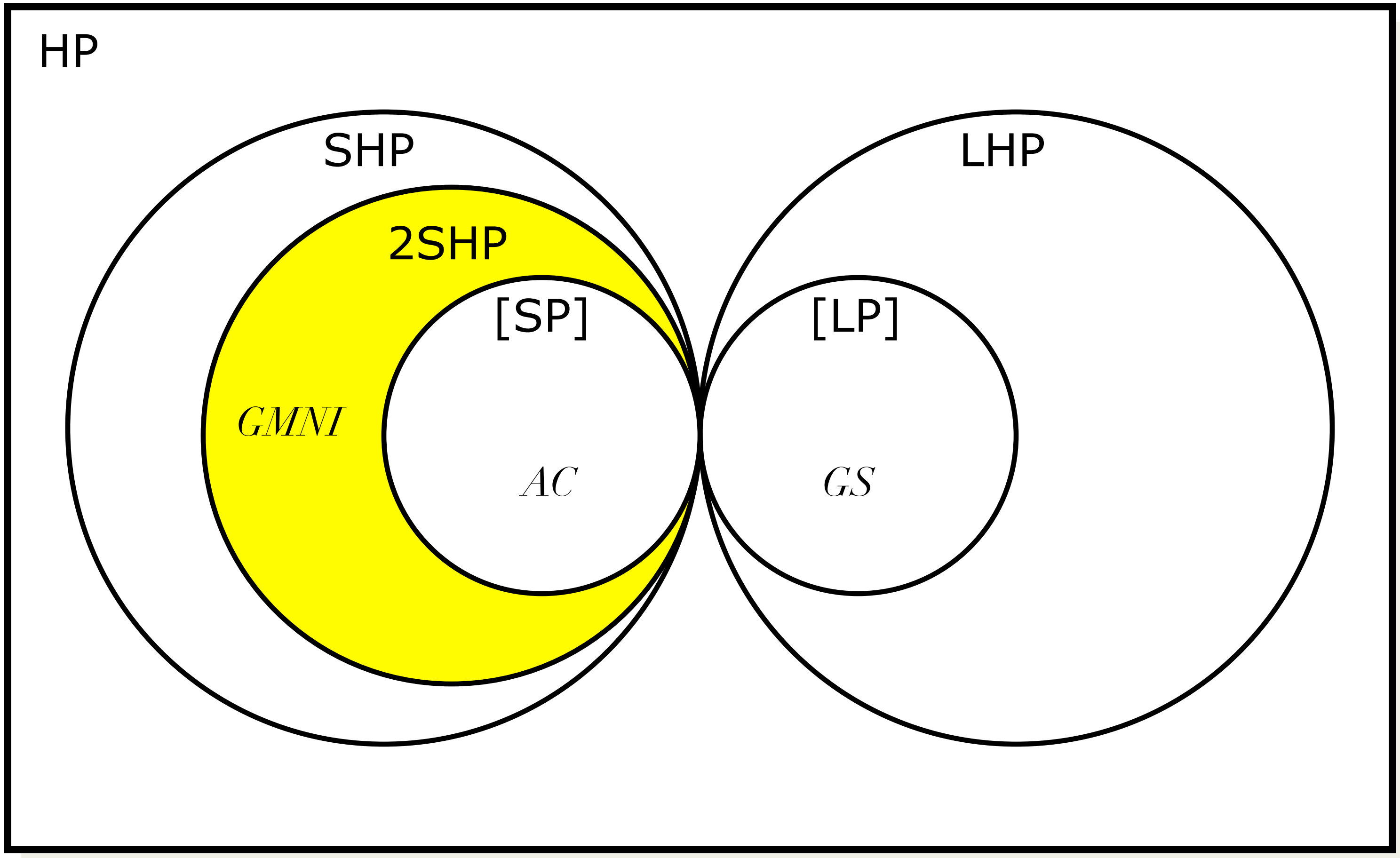
Lifted safety properties [SP]
Lifted liveness properties [LP]



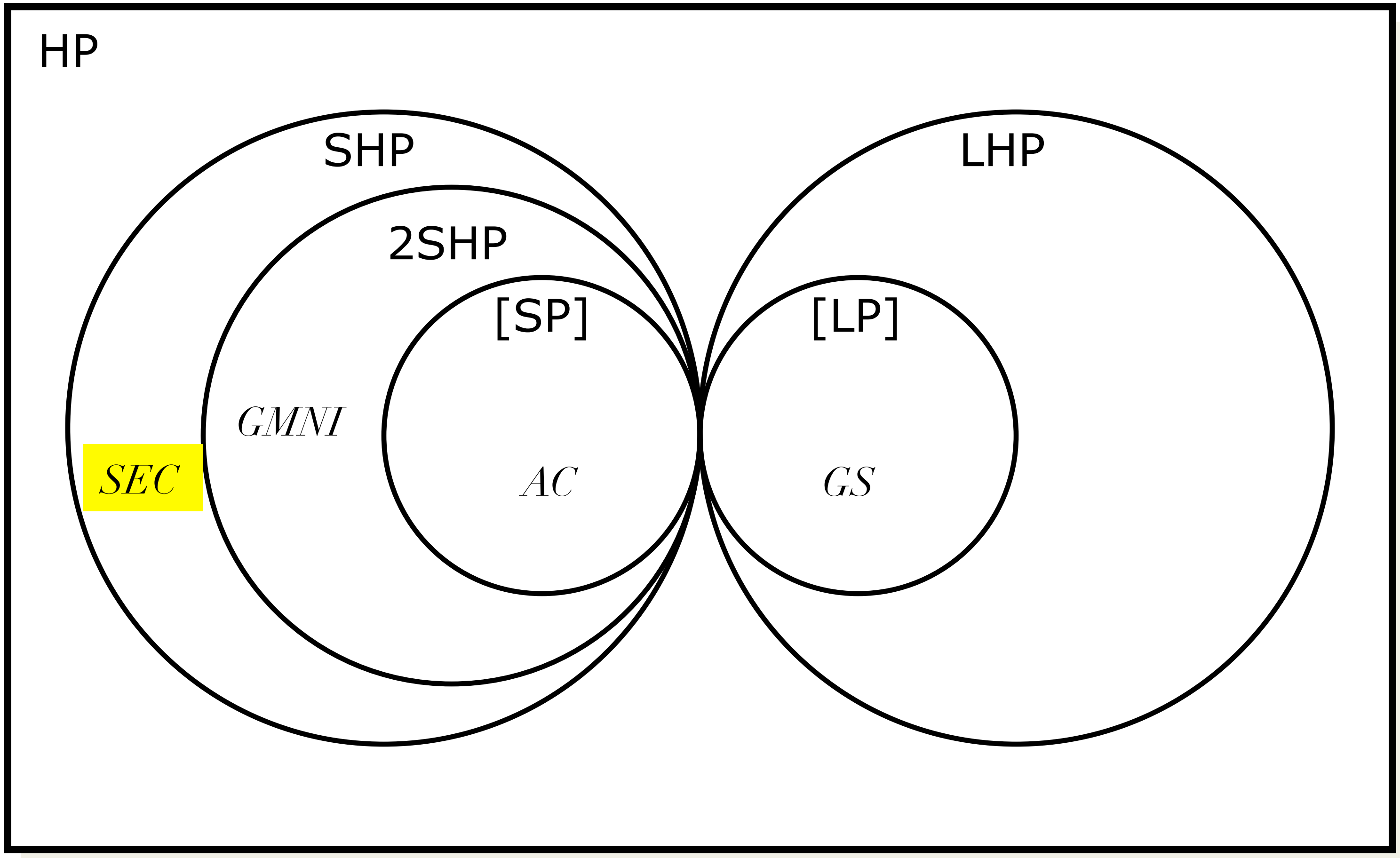
Access control (AC) is safety
Guaranteed service (GS) is liveness



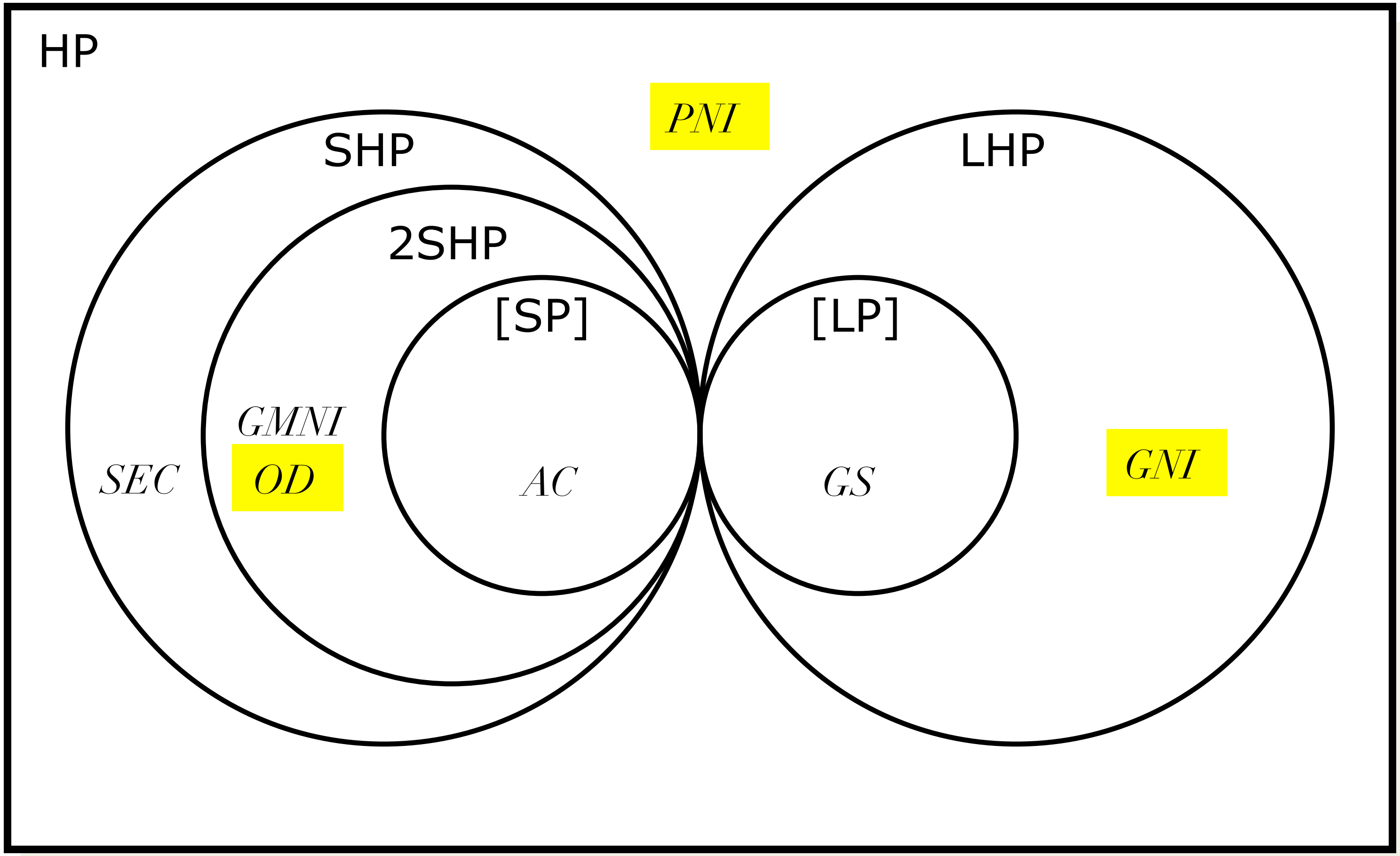
Goguen and Meseguer's noninterference ($GMNI$)
is hypersafety



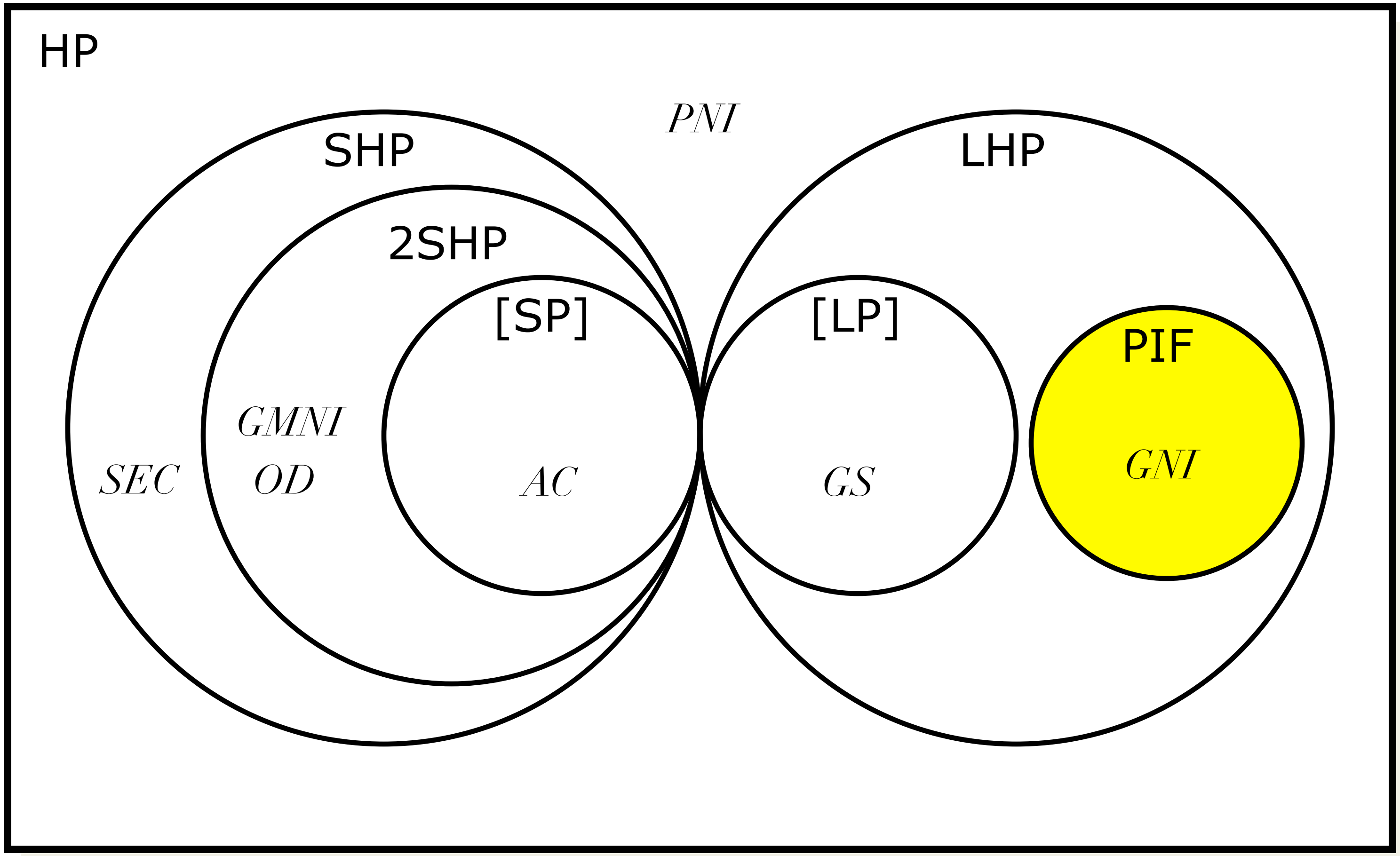
2-safety hyperproperties (2SHP)



Secret sharing (SEC) is not *k*-hypersafety for any *k*



Observational determinism (OD) is 2-hypersafety
Generalized noninterference (GNI) is hyperliveness
Probabilistic noninterference (PNI) is neither



Possibilistic information flow (PIF) is hyperliveness

Revisiting the CIA Landscape

- **Confidentiality**
  - Information flow is not a property
  - Is a hyperproperty (HS: OD; HL: GNI)
- **Integrity**
  - Safety property?
  - Dual to confidentiality, thus hyperproperty?
- **Availability**
  - Sometimes a property (max. response time)
  - Sometimes a hyperproperty (HS: % uptime, HL: avg. resp. time)

→ CIA seems unrelated to hyperproperties

Reading

- **Hyperproperties.** *Journal of Computer Security* 18(6): 1157–1210, 2010. With Fred B. Schneider.
- **Temporal Logics for Hyperproperties.** In *Proc. POST*, pp. 265-284, 2015. With Bernd Finkbeiner, Masoud Koleini, Kristopher Micinski, Markus Rabe, and Cesar Sanchez.

Upcoming events

- [May 16] Final exam