Review

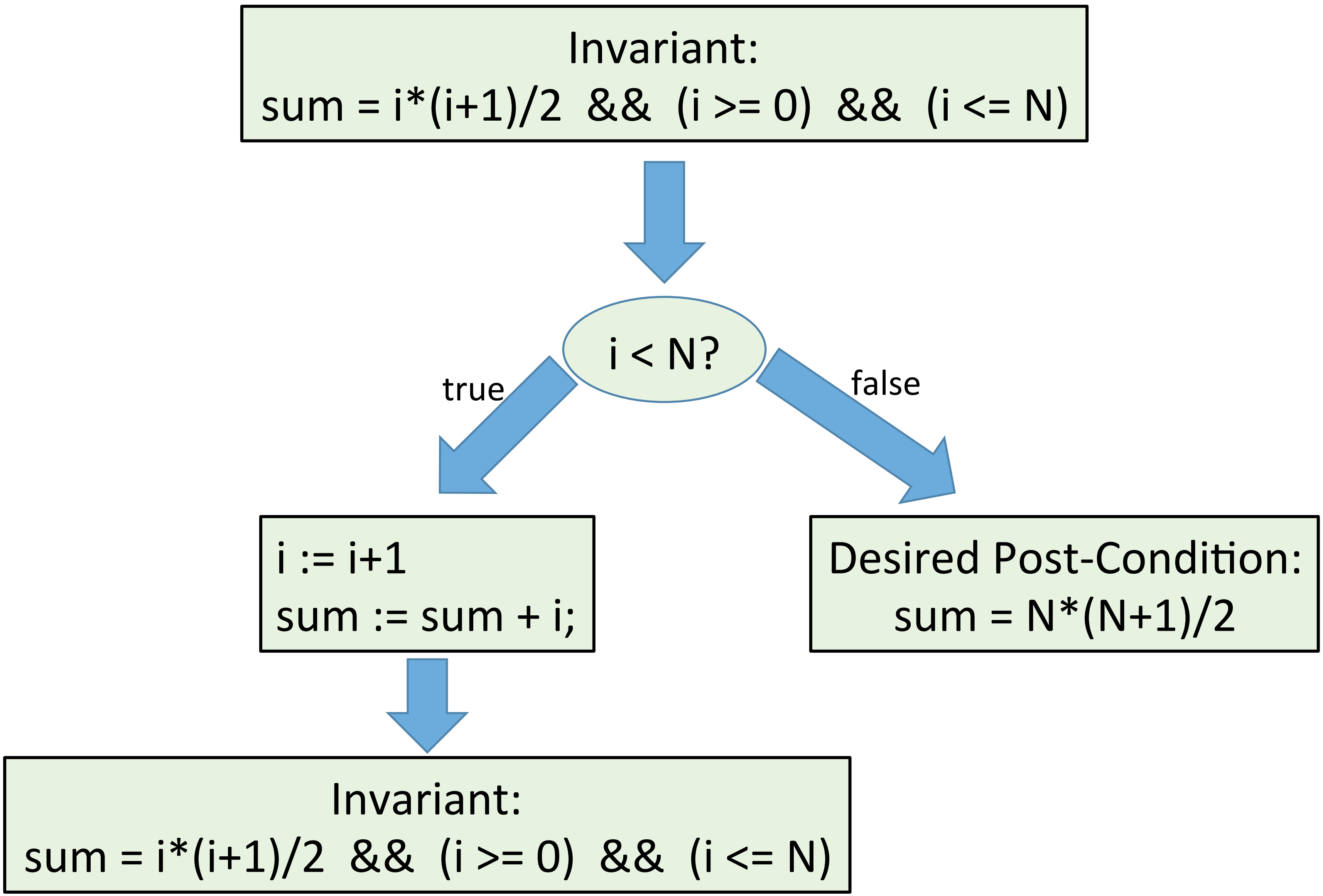
Program Verification => Satisfiability

```
method Eg1 (x, y, z: bool)
{
   var result : bool;
   if (x)
      result := y;
   else
      result := z;
   assert result;
}
```

(x,y,z) is a counterexample iff
$(\neg x \lor \neg y) \land (x \lor \neg z)$

Demo

```
method Sum (N: int)
    returns (sum : int)
    requires N > 0;
    ensures sum == N*(N+1)/2;
{
    var i := 0; sum := 0;
    while (i < N)
       invariant (sum == i*(i+1)/2)
          && (i >= 0) && (i <= N)
    {
       i := i + 1;
       sum := sum + i;
    }
}
```


Invariant:
sum = i*(i+1)/2 && (i >= 0) && (i <= N)
i < N?
true
false
i := i+1
sum := sum + i;
Desired Post-Condition:
sum = N*(N+1)/2
Invariant:
sum = i*(i+1)/2 && (i >= 0) && (i <= N)

Review

Propositional Logic

<u>Syntax</u>

A formal language for expressing some class of assertions

<u>Semantics</u>

What do we mean by these assertions?

- M is a model for ϕ
  - $M \vDash \phi$
- ϕ is satisfiable
- ϕ is a tautology
  - $\vDash \phi$

Propositional Satisfiability

- How can we check if
  - ϕ is a tautology?
  - ϕ is satisfiable?

- Decidable
  - Only finitely many cases to check

- Efficiency?
  - Original NP-Complete problem
  - But very good SAT solvers have been developed over the years …

Syntax

A formal language
for expressing
some class of assertions

Semantics

What do we mean
by these assertions?

Proofs & Proof Systems

What constitutes a
valid proof
of an assertion?

$M \models \phi$

$\models \phi$

Formal Proofs & Proof Systems

- Exhaustive checking does not work, e.g., when we reason about integers:
  - For all x,y,z, $(x<y) \wedge (z=x+y/2) \Rightarrow (z<y)$

- Need other approaches to proofs

- Goal: Finite reasoning about infinitely many possibilities

First Order Logic
aka
Predicate Calculus

First Order Logic

Example

- Consider whole numbers (the universe)

- Assertions are written using the symbols
  - $0, 1$: constant symbols
  - $+$ (addition), $\times$ (multiplication): function symbols
  - $\leq$: predicate symbols
  - x, y, z : (logical) variables

  - $\exists$: existential quantification
  - $\forall$: universal quantification

First Order Logic
(Informal) Semantics

- Existential Quantification
  - $\exists x.\ \varphi(x)$
  - There exists some element x (in the universe) such that $\varphi(x)$ holds

- Universal Quantification
  - $\forall x.\ \varphi(x)$
  - For every element x (in the universe) $\varphi(x)$ holds

First Order Logic
Example

- Consider the natural numbers
  - Let $\times$ denote multiplication

- What does the following say?
  - $\exists z.\ x \times z = y$
  - y is a multiple of x
  - In this assertion (formula), z is a bound variable and x and y are free (unbound) variables

- What does the following say?
  - $\forall x \forall y \quad x \times y = z \;\Rightarrow\; (x=1) \vee (x=z)$
  - z is a prime number

First Order Logic:

Propositional Logic +

- Variables: $x, y, z, \ldots$
- Function symbols: $f, g, +, \times, \cdot$
  - arity: number of operands
  - prefix notation: $f(x,y)$
  - infix notation: $x+y$
  - constant symbols: $0, 1, \ldots$
- Predicate symbols: $p, q, >, \geq$
  - Equality predicate: $\mathrm{x}=\mathrm{y}$ (Predefined “predicate” with a fixed meaning/interpretation)
- Quantification (Universal/Existential)

First Order Logic: Syntax

- The set of terms is defined by:

$$\tau ::= f(\tau_1 \cdots, \tau_n) \mid x$$

- Examples: $x+1$, $x \times (y+z)$

- The set of formula is defined by:

$$\phi ::= p(\tau_1, \cdots, \tau_n) \mid \tau_1 = \tau_2 \mid$$

$$\neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \forall x.\, \phi \mid \exists x.\, \phi$$

- Examples: $x \geq y+z$, $\forall x \forall y\ (x \geq y) \wedge (y \geq x) \Rightarrow (x = y)$

- A sentence is a formula with no free variables

First Order Logic
Example

- Consider set theory
  - $\in$: predicate symbol

- What does the following say?
  - $\forall z.\ z \in x \Rightarrow z \in y$
  - "x is a subset of y"

- What does the following say?
  - $\forall w.\ (w \in z) \Leftrightarrow\ (w \in x) \vee (w \in y)$
  - "z is the union of x and y"

Examples

- Natural numbers (Peano arithmetic)
  - Constant symbol: 0
  - Function symbol: S (successor function)

- Natural numbers:
  - Constant symbol: 0
  - Function symbol: S (successor function)
  - Function symbols: $+, \times$

- Set theory
  - Constant symbol: ϕ (optional)
  - Predicate symbol: $\in$

First Order Logic

Quantification: Exercise

- What's the difference between:
  - $\forall x.\ \exists y.\ (x \leq y)$
  - $\exists y.\ \forall x.\ (x \leq y)$
- Conversions between $\exists$ and $\forall$
  - $\neg\exists x.\ \phi(x)$ equivalent to $\forall x.\ \neg\phi(x)$
  - $\neg\forall x.\ \phi(x)$ equivalent to $\exists x.\ \neg\phi(x)$

First Order Logic
More Exercises

- What do the following mean?
  a) $\exists x \forall y \; x \oplus y = y$
  b) $\exists x \forall y \; (x \oplus y = y) \wedge (y \oplus x = y)$
  c) $\forall x \forall y \; x \oplus y = y \oplus x$
- Does (a) hold
  - If we consider the set of integers and interpret $\oplus$ as integer-addition?
- Find an example of a set and an operation $\oplus$ that does not satisfy (a)

First Order Logic: Semantics

- We can interpret terms and formulae …
- … given the *meaning* of the function symbols and predicate symbols
  - A set A (the universe)
  - For every function-symbol f of arity n, a function $M[f] : A \uparrow n \rightarrow A$ representing the interpretation of f
  - For every predicate-symbol p of arity n, a function $M[p]: A \uparrow n \rightarrow \{T,F\}$ representing the interpretation of p
  - (called a *structure* or *interpretation* for the underlying language)
  - We will refer to the structure as M

First Order Logic: Semantics

- Extend the interpretation-function to define the value $M[\tau] \in A$ for any term τ inductively.
- Extend this to evaluate any sentence φ as being true of false in M.
- We write $M \vDash \phi$ to denote that ϕ holds true in the interpretation M.
- We define $M \vDash \phi$ inductively.

Mathematical Preliminaries

Inductive Definitions

- Syntax $\phi ::= P \quad | \quad \phi_1 \vee \phi_2 \quad | \quad \phi_1 \wedge \phi_2$

Mathematical Preliminaries

Inductive Definitions

- Let $\Sigma = \mathcal{P} \cup \{\wedge, \vee, \neg\}$
- Let Σ^* denote the set of all sequences of symbols from Σ

- The set of formulas is the smallest subset S of Σ^* that satisfies:
  - If $x \in \mathcal{P}$, then $x \in \mathcal{S}$
  - If $\phi_1 \in \mathcal{S}$ and $\phi_2 \in \mathcal{S}$ then $\phi_1 \vee \phi_2 \in \mathcal{S}$
  - If $\phi_1 \in \mathcal{S}$ and $\phi_2 \in \mathcal{S}$ then $\phi_1 \wedge \phi_2 \in \mathcal{S}$

Mathematical Preliminaries

Inductive Definitions

- Syntax

$$\phi ::= P \quad | \quad \phi\downarrow 1 \vee \phi\downarrow 2 \quad | \quad \phi\downarrow 1 \wedge \phi\downarrow 2$$

Antecedent

$$M \vDash \phi\downarrow 1 , \qquad M \vDash \phi\downarrow 2 \ / M \vDash \phi\downarrow 1 \wedge \phi\downarrow 2$$

- Semantics

Consequent

Mathematical Preliminaries

Inductive Definitions

- Syntax

$$\phi ::= P \quad | \quad \phi_1 \vee \phi_2 \quad | \quad \phi_1 \wedge \phi_2$$

Antecedent

$$\frac{M \models \phi_1 , \qquad M \models \phi_2}{M \models \phi_1 \wedge \phi_2}$$

- Semantics

Consequent

- Similarly for
  - Proof rules
  - Type systems

Example

- Consider the language with
  - function symbols $\oplus$ and $\otimes$ of arity 2, and
  - function (constant) symbols $c\downarrow 0$ and $c\downarrow 1$ of arity 0

- Let M denote the following structure
  - The universe is the set of integers
  - $M[\oplus]$ is integer-addition
  - $M[\otimes]$ is integer-multiplication
  - $M[c\downarrow 0]$ is 0
  - $M[c\downarrow 1]$ is 1

Example

- Does $M \models \neg \exists x.\ (x \otimes x) \oplus c{\downarrow}1 = c{\downarrow}0$ hold?

- Is there any structure N such that $N \models \exists x.\ (x \otimes x) \oplus c{\downarrow}1 = c{\downarrow}0$

Semantic Concepts

- $\mathcal{M}$ is said to be a model for ϕ iff $\mathcal{M} \vDash \phi$
- We say M is a model of a set $\{\psi_1, \psi_2, \cdots\}$ if M is a model of every ψ_i in the set
- ϕ is said to be satisfiable if it has a model
- ϕ is said to be unsatisfiable if it has no model
- ϕ is said to be valid (or a tautology) if every interpretation $\mathcal{M}$ is a model for ϕ
- We write $\vDash \phi$ iff ϕ is a tautology

Syntax

A formal language
for expressing
some class of assertions

Semantics

What do we mean
by these assertions?

Proofs & Proof Systems

What constitutes a
valid proof
of an assertion?

$M \models \phi$

$\models \phi$

Axiomatic Reasoning

- Consider the language (of group theory)
  - one nullary function symbol e
  - one unary function symbol I ($I(a)$ denotes a^{-1})
  - one binary function symbol $\oplus$

- Consider the following "axioms":
  - A_1 : $\forall x \forall y \forall z.\ x \oplus (y \oplus z) = (x \oplus y) \oplus z$
  - A_2 : $\forall x.\ e \oplus x = x$
  - A_3 : $\forall x.\ I(x) \oplus x = e$
  - A_2' : $\forall x.\ x \oplus e = x$
  - A_3': $\forall x.\ x \oplus I(x) = e$

Example

- Let ϕ denote the formula $\forall x \forall y \forall z.\ (x \oplus y = x \oplus z) \Rightarrow y = z$
- What does ϕ say?
- Let M be a structure such that
  - $M \vDash A{\downarrow}1$
  - $M \vDash A{\downarrow}2$
  - $M \vDash A{\downarrow}3$
- Does $M \vDash \phi$ hold?

Axiomatization

- We write $\{A_1, A_2, A_3\} \vDash \varphi$ to mean that
  - Every model of $\{A_1, A_2, A_3\}$ is a model of φ
  - I.e., if M is any structure such that $M \vDash A_1$, and $M \vDash A_2$ and $M \vDash A_3$ then $M \vDash \varphi$.

- Let Ψ be a set of formula (axioms or axiom schemas)

- We write $\Psi \vDash \varphi$ to mean that
  - Every model of Ψ is a model of φ
  - Thus, φ is a semantic consequence of Ψ
  - A semantic concept … no easy way to check.

- The theory of Ψ is the set of all φ such that $\Psi \vDash \varphi$

Axiomatization

- Suppose we "axiomatize" M using a set Ψ of formula (axioms)
  - That is, $M \vDash \psi$ for every $\psi \in \Psi$
  - That is, M is a model of Ψ

- Problem reduction:

Does $M \vDash \phi$?

$\Leftarrow$

$\Rightarrow$?

Does $\Psi \vDash \phi$?

Theory Completeness

- For every φ (with no free variables)
  - Either $M \vDash \varphi$ or $M \vDash \neg\varphi$
  - It is possible that neither $\Psi \vDash \varphi$ nor $\Psi \vDash \neg\varphi$

- We say that Ψ is complete (or the theory of Ψ is complete) if
  - for every φ either $\Psi \vDash \varphi$ or $\Psi \vDash \neg\varphi$

Syntax

A formal language
for expressing
some class of assertions

Semantics

What do we mean
by these assertions?

Proofs & Proof Systems

What constitutes a
valid proof
of an assertion?

$M \vDash \phi$

(reduction 1)

$\Psi \vDash \phi$

$\vDash \phi$

(reduction 2)

$\Psi \vdash \phi$

$\vdash \phi$

Proofs & Proof Systems

- A proof system (or deduction system) is used to define what a valid proof is

- A proof is a tree-like structure
  - Leafs: axioms (or axiom instances)
  - Internal nodes: compose sub-proofs using inference rules
  - Root: the theorem that is proven
  - (convenient to draw upside-down)

Proofs & Proof Systems

- A proof-system $\mathcal{S}$ is an inductive definition of judgements of the form $\vdash_{\mathcal{S}} \phi$ or $\Psi \vdash_{\mathcal{S}} \phi$
- We use the judgement $\vdash_{\mathcal{S}} \phi$ to denote that ϕ can be proven to be valid (in system $\mathcal{S}$)
- The judgement $\Psi \vdash \phi$ denotes that ϕ can be proven given proofs of all $\psi \in \Psi$ (in system $\mathcal{S}$).

Example

$$/\Psi, \phi \vdash \phi$$

$$\Psi \vdash \phi_1 , \qquad \Psi \vdash \phi_1 \Rightarrow \phi_2 \; / \Psi \vdash \phi_2$$

(modus ponens)

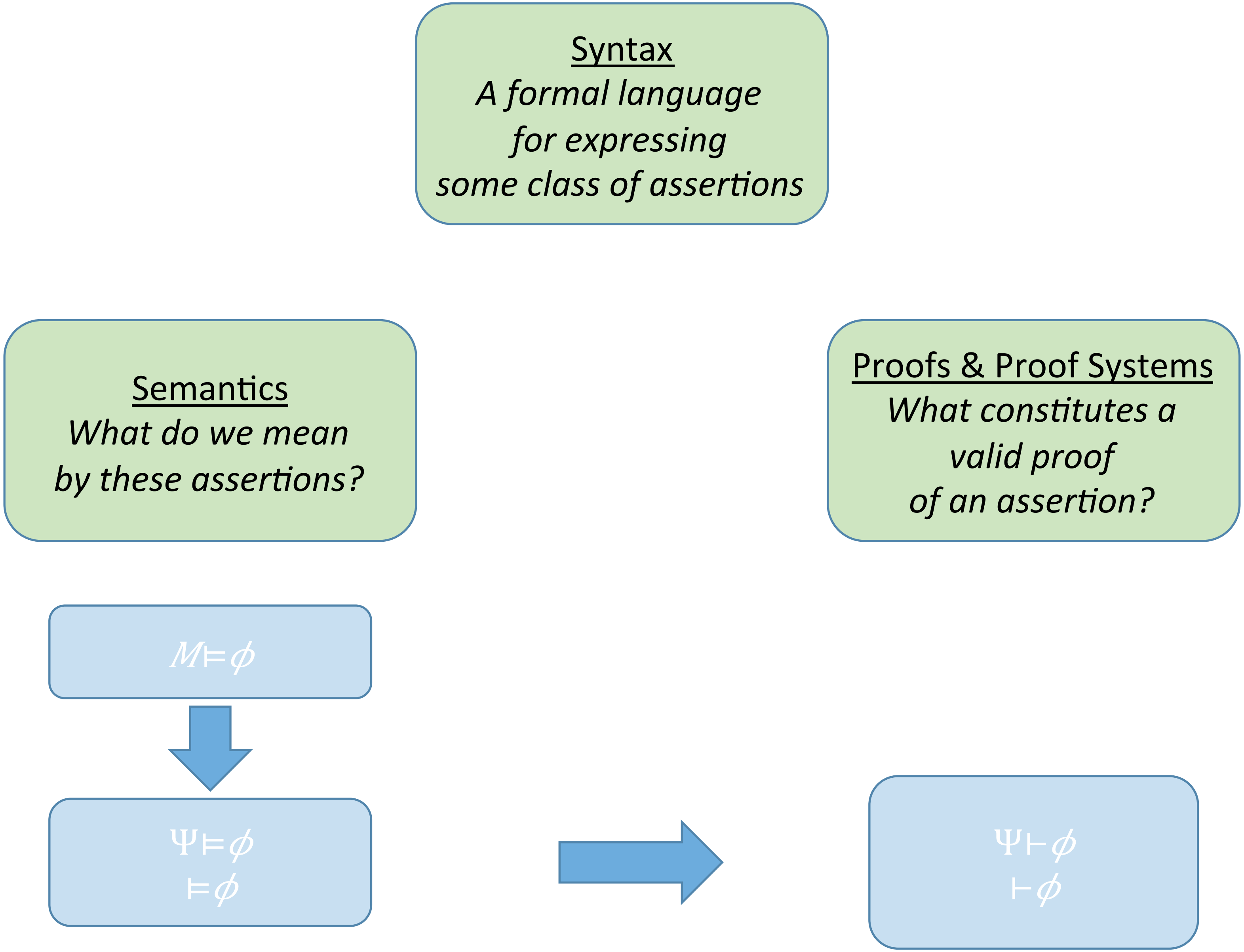
$$\Psi, \phi_1 \vdash \phi_2 \; / \Psi \vdash \phi_1 \Rightarrow \phi_2$$

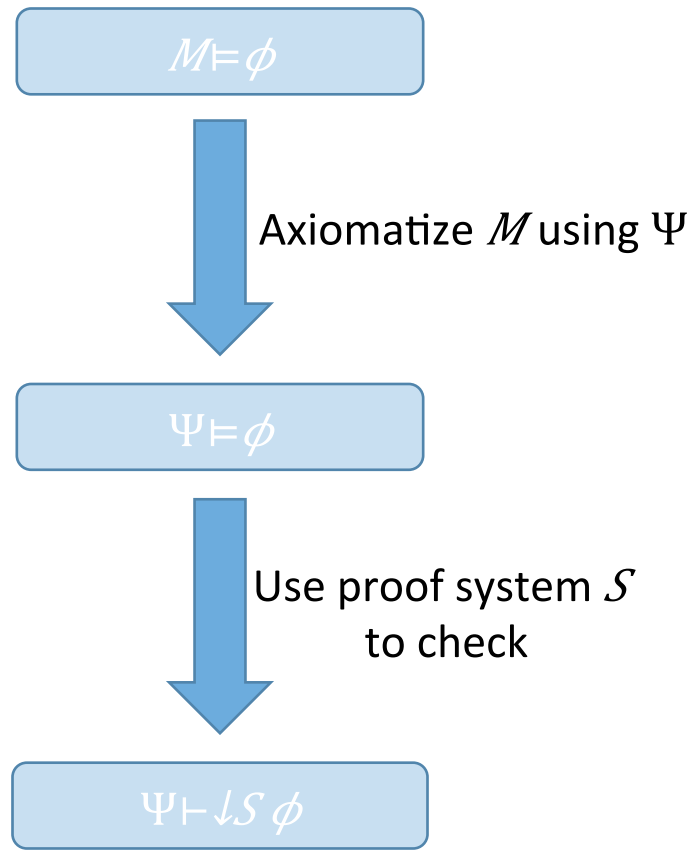
$$\Psi \vdash \phi_1 , \qquad \Psi \vdash \phi_2 \; / \Psi \vdash \phi_1 \wedge \phi_2$$

Soundness & Completeness

- A proof system is said to be sound if all provable formulae are valid: that is,
  - $\Psi \vdash \phi$ implies $\Psi \vDash \phi$

- A proof system is said to be complete if all valid formulae are provable: that is,
  - $\Psi \vDash \phi$ implies $\Psi \vdash \phi$


Syntax
A formal language
for expressing
some class of assertions
Semantics
What do we mean
by these assertions?
Proofs & Proof Systems
What constitutes a
valid proof
of an assertion?
$M \vDash \phi$
$\Psi \vDash \phi$
$\vDash \phi$
$\Psi \vdash \phi$
$\vdash \phi$



$M \vDash \phi$
Axiomatize M using Ψ
$\Psi \vDash \phi$
Use proof system $\mathcal{S}$
to check
$\Psi \vdash_{\mathcal{S}} \phi$

Godel’s Completeness & Incompleteness Theorems

Summary

- By design [of formal proof systems]
  - Correctness of a given proof can be easily machine-checked
    - But can be tedious for us to write
  - The set of proofs (for a chosen set of axioms) is recursively enumerable
    - Can automate search for proofs
    - Challenges
      - Efficiency
      - Choosing a set of axioms

Satisfiability Modulo Theories (SMT Solvers)

- Extend SAT solvers to check satisfiability modulo one or more theories

```
method Eg2 (x, y : int)
  returns (z : int)
{
   assume x < y;
   z := (x+y)/2;
   assert x < z;
}
```

Valid iff for all x, y, z:
$((x<y)\wedge(z=x+y/2\ )\Rightarrow(x<z)\ )$

(x,y,z) is a counterexample iff
$(x<y)\wedge(z=x+y/2\ ))\wedge(x\geq z)$