Logic for Verification

João Martins

August 4, 2012

1. Introduction
2. Syntax
3. Semantics
4. Verification
    - Truth Tables
    - Resolution
    - Natural Deduction
5. Beyond Propositional Logic
    - First Order Logic
    - Modal Logics
    - Dynamic Logic
    - Hybrid systems and Differential Dynamic Logic

Summary

1. Introduction
2. Syntax
3. Semantics
4. Verification
    - Truth Tables
    - Resolution
    - Natural Deduction
5. Beyond Propositional Logic
    - First Order Logic
    - Modal Logics
    - Dynamic Logic
    - Hybrid systems and Differential Dynamic Logic

What is logic?

- A naturally human process that allows us to reason about truth
- Language with specific symbols (syntax) that are given meaning (semantics)
- Usually accompanied with techniques to check the validity of its assertions
- It can be a powerful (computational) tool to derive consequences from hypotheses

Helps YOU think more rigorously! :)

Assertions: examples

- It is getting late and we are still in school
- If John doesn't catch the bus, he'll be late
- Either Mary's at the movies or John is home and Brian is sleeping
- Peter went to the doctor and got sick

Deduction

What is deduction?

- Rules that tell you what else is true given certain premises

Some examples of deduction:

- If we assume *a* and reach a contradiction, then *a* must be false
- If *a* is true or *b* is true and from either we can derive *c*, then *c* must also be true
- If we know that *a* implies *b*, and we know that *a* is true, then *b* must be true (*modus ponens*)
- If we know that *a* implies *b*, and we know that *b* is false, then we know that *a* must also be false (*modus tollens*)
- Etc...

Propositional logic

Propositional logic:

- Talks about... propositions (surprise!)
- Defines the behaviour of basic logic connectives $(\wedge, \rightarrow, \neg, ...)$

Propositions, typically p or q can stand for “it’s raining”, or “logic is interesting”.

Propositions can either be true or false, and more complex *formulae* can be constructed from the connectives and the propositions.

Summary

1. Introduction
2. Syntax
3. Semantics
4. Verification
   - Truth Tables
   - Resolution
   - Natural Deduction
5. Beyond Propositional Logic
   - First Order Logic
   - Modal Logics
   - Dynamic Logic
   - Hybrid systems and Differential Dynamic Logic

What is syntax?

- Syntax restricts what sequences of symbols and propositions we may write.
- Syntax does not say anything about their meaning.
- A symbol $\perp$ for falsity/absurdity and the elements of a set of propositions P are called *atomic formulae*.
- The "good" sequences of symbols and propositions are called *formulae*.

Connectives and natural language

Propositional logic has the following connectives:

- $\vee$, or *disjunction*, is an alternative. $a \vee b$ is read as "*a* is true or *b* is true", or "at least one of *a* and *b* must be true"
- $\wedge$, or *conjunction*, indicates that both parts must be true. $a \wedge b$ represents the fact that both *a* and *b* must be true.
- $\rightarrow$ represents the notion of consequence, with $a \rightarrow b$ being read as "if *a* then *b*", "*b* if *a*", "*a* only if *b*", etc...

Actual syntax for propositional logic

Definition (Propositional language induced by a set of symbols)

Let P be a set of propositions. Then the propositional language F_P induced by P is inductively given by (or the smallest set such that):

- $\bot \in F_P$
- If $p \in P$, then $p \in F_P$
- If $A, B \in F_P$ then $(A \vee B) \in F_P$, $(A \wedge B) \in F_P$ and $(A \rightarrow B) \in F_P$

Examples

- “I like logic!” can be written p
- “I like math!” can be written q
- “I like logic and math!” can be written $p \wedge q$
- “I like math, therefore I like logic!” can be written $q \rightarrow p$

Exercises!

Exercise

1. Peter went to the doctor and got sick.
2. Peter is home sick
3. Peter is sick if he has the flue
4. Peter does not have flue if he does not have fever
5. Peter is home because he got sick
6. Peter stays home only if he is sick
7. Peter got sick, but has already been to the doctor
8. Being sick or going to the doctor make Peter annoying
9. If Peter went to the doctor because he is sick, then he's not home
10. Peter goes to the doctor if he's sick and Hannah is bored, unless the weather is bad

Syntactic sugar

- Negation: $\neg A \triangleq A \to \bot$, or "$A$ is false" is defined by A implying the absurd
- Truth: $\top \triangleq \neg\bot$
- Equivalence: $A \leftrightarrow B \triangleq (A \to B) \wedge (B \to A)$

Some examples:

- $\neg(\neg A \wedge B)$ is $((A \to \bot) \wedge B) \to \bot$
- $\neg A \leftrightarrow (B \vee C)$ is $((A \to \bot) \to (B \vee C)) \wedge ((B \vee C) \to (A \to \bot))$

Summary

1. Introduction
2. Syntax
3. Semantics
4. Verification
   - Truth Tables
   - Resolution
   - Natural Deduction
5. Beyond Propositional Logic
   - First Order Logic
   - Modal Logics
   - Dynamic Logic
   - Hybrid systems and Differential Dynamic Logic

What are semantics?

- Semantics assigns a *meaning* to purely syntactic symbols
- It enables us to give propositions a truth value (true or false)
- Tells us the truth value of formulae from the truth value of propositions *and* the meaning of the connectives

For example,

- $a + b$ could that at least one of a and b must be true.
- $a * b$ could be that both a and b must be true.
- $a \oplus b$ could be that at least and at most one of a and b must be true.

Thus, semantics deals with the *validity* and *satisfaction* of logical formulae.

More specifically...

We want semantics to determine the truth value of a formula.
To do that,

- We must assign truth values to each proposition $p \in P$
- Attach meaning to the connectives
- Evaluate a formula's subformulae, interpreting connectives as a function

Satisfaction of a formula

Definition (Satisfaction)

Let $V : P \to \{0, 1\}$ be a valuation. The satisfaction of a formula A by $V \in F_P$, denoted $V \Vdash A$ is defined inductively as follows:

- $V \Vdash p$ if $V(p) = 1$ $(p \in P)$
- $V \Vdash \bot$ *never* holds
- $V \Vdash A \vee B$ if $V \Vdash A$ or $V \Vdash B$
- $V \Vdash A \wedge B$ if $V \Vdash A$ and $V \Vdash B$
- $V \Vdash A \to B$ if whenever $V \Vdash A$ then $V \Vdash B$

We are giving, in natural language, which we know and understand, the intended meaning to the symbols.

More notation

Notation and terminology

- If $V \Vdash A$, we say A is satisfied by V
- We write $V \nVdash A$ if $V \Vdash A$ does not hold
- $V \Vdash \neg A$ if and only if $V \nVdash A$
- Given $\mathcal{A} \subseteq F_P$, $V \Vdash \mathcal{A}$ if for every $A \in \mathcal{A}$, $V \Vdash A$

$V \Vdash \neg A$ is re-written as $V \Vdash A \to \perp$, which means that if A is false, the implication is true. If A is true, we get the absurd, so it cannot happen.

Possible, contradictory, valid

Terminology

A formula $A \in F_P$ is...

- possible if for some V, $V \Vdash A$
- contradictory if there is no V such that $V \Vdash A$
- valid (denoted $\Vdash A$) if for all V, $V \Vdash A$
- Valid formulae are also called *tautologies*
- We write $\nVdash A$ if A is not a tautology
- $\mathcal{A} \subseteq F_P$ is possible if there exists a V that satisfies all $A \in \mathcal{A}$. Otherwise it is contradictory.

HAH - more exercises!

Exercises

Show, using the definitions, whether the following are possible or contradictory:

1. $a \wedge \neg a$
2. $a \wedge b$
3. $(a \rightarrow b) \wedge (a \wedge \neg b)$

Exercises

Show, using the definitions, the validity of the following:

1. $a \vee \neg a$
2. $a \rightarrow (a \vee b)$
3. $\neg(a \vee b) \rightarrow \neg a$
4. $((a \rightarrow b) \wedge a) \rightarrow b$ (you can also do by absurd)

Semantic consequence

Here is one of the most important notions in logic:

Definition (Semantic Consequence)

Let $\mathcal{A} \subseteq F_P$ and $A \in F_P$. We say that A is semantic consequence of $\mathcal{A}$, denoted $\mathcal{A} \models A$, if for each V, if $V \Vdash \mathcal{A}$, then $V \Vdash A$.

Example

If the subway is late (s) and there are no cabs in the station ($\neg c$), Peter gets home late (l). Peter is not late, but the subway was late. Therefore, there were cabs at the station.

$\{(s \wedge \neg c) \rightarrow l, \neg l \wedge s\} \models c$

Definition (Semantic Equivalence)

Two formulae A and B are said to be logically equivalent, denoted by $A \equiv B$ if we have $\{A\} \models B$ if and only if $\{B\} \models A$.

Even more exercises, sorry guys :(

Exercises

Check whether the following are true or false:

1. $\{\neg(a \wedge b), a\} \models \neg b$
2. $\{\neg(a \rightarrow b), \neg b\} \models \neg a$
3. $\{a \rightarrow b, \neg a \rightarrow b\} \models b$
4. $\{a \rightarrow b\} \models (a \wedge c) \rightarrow b$
5. $\{(a \wedge b) \rightarrow c, d \rightarrow a\} \models b \rightarrow (d \rightarrow c)$

Pro-tip: if you have tons of implications, using *reductio ad absurdum* may turn them into ands! *hint hint*

Some more cute details

Are these true?

- $\{A \wedge B\} \models A$
- $\{A\} \models A \vee B$
- $\{\bot\} \models A$ (why?)

Proposition

$\{A\} \models B$ iff $\Vdash A \rightarrow B$.

Proof: Let's show the $\Rightarrow$ direction first. By hypothesis, $\{A\} \models B$, by definition is for any V, if $V \Vdash A$ then $V \Vdash B$. Again by definition, that is exactly $\Vdash A \rightarrow B$.

The $\Leftarrow$ direction is similar.

Some shortcuts!

More (provable) laws of propositional logic

- Double-negation: $\neg\neg A \equiv A$
- Contradiction: $A \wedge \neg A \equiv \bot$
- de Morgan Laws:
  - $\neg(A \wedge B) \equiv \neg A \vee \neg B$
  - $\neg(A \vee B) \equiv \neg A \wedge \neg B$
- Distributivity:
  - $A \rightarrow (B \rightarrow C) \equiv (A \rightarrow B) \rightarrow (A \rightarrow C)$
  - $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$
  - $(A \vee B) \wedge C \equiv (A \vee C) \wedge (B \vee C)$
  - $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$
  - $(A \wedge B) \vee C) \equiv (A \wedge C) \vee (B \wedge C)$

Do we need all connectives?

The de Morgan laws tell us $A \wedge B \equiv \neg(\neg A \vee \neg B)$. We don't *need* $\wedge$ if we have $\neg$ and $\vee$.

Exercise

Define $\neg$, $\vee$, $\wedge$ and $\leftrightarrow$ from $\perp$ and $\rightarrow$

Exercise

Define $\perp$, $\wedge$, $\rightarrow$ and $\leftrightarrow$ from $\neg$ and $\vee$

One of you asked: how do we know the logic is consistent?

Plus, you said you wanted more math :D

Proposition

Let V_1 and V_2 be two valuations over P. For any $A \in F_P$, if $V_1(a) = V_2(a)$ for every $a \in P$, then $V_1 \Vdash A$ if and only if $V_2 \Vdash A$.

One of you asked: how do we know the logic is consistent?

Proof

By induction on the formula A. **Base case:**

- $A = \bot$, trivially $V_1 \Vdash \bot$ iff $V_2 \Vdash \bot$ since for all V, $V \nVdash \bot$
- $A = a$, $a \in P$. By hypothesis, $V_1(a) = V_2(a)$, so it follows trivially that $V_1 \Vdash a$ iff $V_2 \Vdash a$

Induction step:

- $A = A_1 \rightarrow A_2$: is $V_1 \Vdash A_1 \rightarrow A_2$ iff $V_2 \Vdash A_1 \rightarrow A_2$? Since A_1 and A_2 are subformulae, by induction hypothesis we have that $V_1 \Vdash A_1$ iff $V_2 \Vdash A_1$ and similarly for A_2. Then, the truth value of $A_1 \rightarrow A_2$ is defined by whatever value A_1 and A_2 take, which is the same for V_1 and V_2. □

Exercise :P

Do the case for $A = A_1 \vee A_2$.

Substitution theorem

Substitution Theorem

Suppose $A \equiv B$, and that C has A as a subformula. Let C' be obtained by substitution A for B in C. Then, $C \equiv C'$.

Substitution theorem

Substitution Theorem

Suppose $A \equiv B$, and that C has A as a subformula. Let C' be obtained by substitution A for B in C. Then, $C \equiv C'$.

Proof by induction

Base case:

- $C = p$, for $p \in P$. The only subformula of C is C, and therefore $A = C = p$ and also $B = C'$. By hypothesis, $A \equiv B$, so that $C \equiv C'$.
- $C = \bot$, trivial, as before.

Induction step:

- $C = C_1 \vee C_2$ (other cases similar). By induction, $C_i \equiv C_i'$. By hypotesis, A is subformula of C, there are 3 cases:
  - $A = C$, which is proven like the base cases.
  - $A = C_1$. Then, $C' = C_1' \vee C_2$, from which we conclude $C \equiv C'$.
  - $A = C_2$, same as above

What do we know so far?

- How to construct a logical language inductively
- Connectives, propositions and formulae as syntactic objects
- Valuation as a structure capable of assigning truth values to syntactic objects
- The notion of semantic consequence, or of how we can deduce something from hypothesis
- Some neat properties of propositional logic

This does not help us for verification! How can we automatise this process?

Summary

1 Introduction

2 Syntax

3 Semantics

4 Verification

- Truth Tables
- Resolution
- Natural Deduction

5 Beyond Propositional Logic

- First Order Logic
- Modal Logics
- Dynamic Logic
- Hybrid systems and Differential Dynamic Logic

Verification

- The computer doesn't understand natural language
- The computer doesn't understand *semantics*
- The computer plays with symbols. It is syntactic!
- Wanted: purely syntactic techniques for checking semantic consequence/validity

Summary

1. Introduction
2. Syntax
3. Semantics
4. Verification
   - Truth Tables
   - Resolution
   - Natural Deduction
5. Beyond Propositional Logic
   - First Order Logic
   - Modal Logics
   - Dynamic Logic
   - Hybrid systems and Differential Dynamic Logic

Truth Tables

- Extremely simple way to check the validity of a formula A
- Just lay down a table with all possible truth values for the propositions in A
- Each column contains a subformula of A
- Start with the smallest subfurmulae and fill in the blanks...

Example

Example

Simple example If the subway is late (s) and there are no cabs in the station ($\neg c$), Peter gets home late (l).

$$(s \wedge \neg c) \rightarrow l$$

s	c	l	$\neg c$	$s \wedge \neg c$	$(s \wedge \neg c) \rightarrow l$
0	0	0	X	X	X
0	0	1	X	X	X
0	1	0	X	X	X
0	1	1	X	X	X
1	0	0	X	X	X
1	0	1	X	X	X
1	1	0	X	X	X
1	1	1	X	X	X

Example

Example

Simple example If the subway is late (s) and there are no cabs in the station ($\neg c$), Peter gets home late (l).

$$(s \wedge \neg c) \rightarrow l$$

s	c	l	$\neg c$	$s \wedge \neg c$	$(s \wedge \neg c) \rightarrow l$
0	0	0	1	X	X
0	0	1	1	X	X
0	1	0	0	X	X
0	1	1	0	X	X
1	0	0	1	X	X
1	0	1	1	X	X
1	1	0	0	X	X
1	1	1	0	X	X

Example

Example

Simple example If the subway is late (s) and there are no cabs in the station ($\neg c$), Peter gets home late (l).

$$(s \wedge \neg c) \rightarrow l$$

s	c	l	$\neg c$	$s \wedge \neg c$	$(s \wedge \neg c) \rightarrow l$
0	0	0	1	0	X
0	0	1	1	0	X
0	1	0	0	0	X
0	1	1	0	0	X
1	0	0	1	1	X
1	0	1	1	1	X
1	1	0	0	0	X
1	1	1	0	0	X

Example

Example

Simple example If the subway is late (s) and there are no cabs in the station ($\neg c$), Peter gets home late (l).

$$(s \wedge \neg c) \rightarrow l$$

s	c	l	$\neg c$	$s \wedge \neg c$	$(s \wedge \neg c) \rightarrow l$
0	0	0	1	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	0	0	1

Does it scale?

What if we had 10 friends, and all of them could be late?

Does it scale?

What if we had 10 friends, and all of them could be late?

$2^{10} = 1024$ Welp... this could get boring...

Does it scale?

- What if each friend can choose to wear pants or skirts/kilts
- ... and a shirt or a t-shirt
- That's three variables for each person, whether they are late, wearing pants, or t-shirt.

$$2^{30} = 1,073,741,824$$

One billion, seventy-three million, seven-hundred and forty-one thousand, eight-hundred and twenty four (that's right, I took the time to write this down!)

Volunteers?

Perhaps you could split into 4 groups, you'd only get 250 million each!

Summary

1. Introduction
2. Syntax
3. Semantics
4. Verification
   - Truth Tables
   - Resolution
   - Natural Deduction
5. Beyond Propositional Logic
   - First Order Logic
   - Modal Logics
   - Dynamic Logic
   - Hybrid systems and Differential Dynamic Logic

Conjunctive Normal Form: CNF

- A formula A is in CNF if it is a conjunction of disjunctions of literals
- Wait what?
- A is a literal if it is p or $\neg p$, for any $p \in P$
- $A = (a_{11} \vee ... \vee a_{1n_1}) \wedge ... \wedge (a_{n1} \vee ... \vee a_{nn_n})$

CNF as sets

- We represent $\neg a$ as $\overline{a}$.
- We represent $a \vee \neg b \vee d$ as $\{a\overline{b}d\}$
- We represent $(a \vee \neg b \vee d) \wedge (d) \wedge (\neg d \vee a)$ as $\{a\overline{b}d, d, \overline{d}a\}$

All formulae can be CNF

Lemma

Let $A \in F_P$. Then, there exists $B \in F_P$ such that B is in CNF and $A \equiv B$.

Proof (sketch)

By induction. The base cases are already in CNF. For $A = A_1 \wedge A_2$ we have by I.H. that A_1' and A_2' are in CNF and are equivalent to A_1 and A_2 respectively. Therefore, $A_1' \vee A_2'$ is equivalent to $A_1 \vee A_2$ and is in CNF. For $A = A_1 \vee A_2$ you do the same, but use distributivity to get CNF.

Resolution

- An algorithm for checking the satisfiability of a formula in CNF
- Use the following reasoning:

$$\text{if } a \vee b \text{ and } \neg a \vee c \text{ then } b \vee c$$

- $b \vee c$ is called the *resolvent* of $a \vee b$ and $\neg a \vee c$
- This generalises to larger disjunctions ($\vee$)

Simple examples: you know what comes next, right?:)

Examples

1. $\{ab\bar{a}, a\bar{c}c, qwertyuiopasdfghjklzxcvbnm\bar{d}\}$
2. $\{ab, \bar{a}\bar{b}, c\bar{a}\bar{c}\}$
3. $\{ab\bar{c}, a\bar{b}, \bar{a}, abc\}$ (what is the truth value of an empty disjunction?)

BAM - exercises!

Exercises

1. $\perp \rightarrow a$
2. $(a \wedge b) \vee (\neg a \wedge \neg a)$
3. $(a \rightarrow b) \wedge (a \rightarrow \neg b)$
4. $(a \vee b) \wedge \neg a \wedge (\neg a \wedge \neg b)$

Summary

1. Introduction
2. Syntax
3. Semantics
4. Verification
   - Truth Tables
   - Resolution
   - Natural Deduction
5. Beyond Propositional Logic
   - First Order Logic
   - Modal Logics
   - Dynamic Logic
   - Hybrid systems and Differential Dynamic Logic

What is a proof?

Elements that can be used in a proof:

- *Axioms*, which are true and can always be used
- *Hypotheses*, which one assumes to be true (the $\mathcal{A}$ in $\mathcal{A} \models A$)
- Rules of inference, which allow us to syntactically obtain new truths, called *theorems*

What is a proof (formally)?

- A proof is a sequence of formulae
- The first elements in the sequence are the hypotheses
- All the elements after that are obtained by the application of a deduction rule
- Deduction rules may use previously proven formulae as hypotheses
- The last formula is the desired conclusion

Notation

Let $\{A_1, ..., A_n\}$ be a set of hypotheses and A be the desired conclusion. Then, we write

$$\{A_1, ..., A_n\} \vdash A$$

if from the hypotheses $A_1, ..., A_n$ one can build a proof for A.

Terminology

Terminology

- If one can prove $\{A_1, ..., A_n\} \vdash A$, then one says φ is a consequence of the set of hypotheses
- If one proves $\emptyset \vdash A$, then A is said to be a theorem of the deductive system (denoted $\vdash A$)

This sounds awfully familiar...

$$\{A_1, ..., A_n\} \models A$$

Are they the same?

Soundness and Completeness: super duper importantness

- $\{A_1, ..., A_n\} \vdash A$ is *syntactic*
- $\{A_1, ..., A_n\} \models A$ is *semantic*
- But they should match!

Desired theorem for all deductive systems

$\{A_1, ..., A_n\} \vdash A$ if and only if $\{A_1, ..., A_n\} \models A$

Soundness and Completeness: MOAR super duper importantness

Definition (Soundness):

If you can find a proof, the conclusion must hold semantically! This is the most important thing: You never want a system that deduces wrong things!

$\{A_1, ..., A_n\} \vdash A$ implies $\{A_1, ..., A_n\} \models A$

Definition (Completeness):

If it is true (semantically), then you can find a proof. This is usually much harder, and sometimes you will not get a complete proof system because the logic is so complex.

$\{A_1, ..., A_n\} \models A$ implies $\{A_1, ..., A_n\} \vdash A$

Natural Deduction

- Natural deduction is an intuitive proof system, similar to human throught processes
- It is not the best for use by computers, but it is easy to understand
- It has rules of inference that allow you to *introduce* and *eliminate* each of the connectives

- If the rules "make sense", this may be sound.
- If we cover all connectives, perhaps we will have completeness.

Absurd rule

Absurd rule

$$\dfrac{\begin{matrix}[\neg A]^m \\ \mathcal{D} \\ \bot\end{matrix}}{A}\ \bot, m$$

Conjunction rules

Conjunction rules

$$\dfrac{\begin{matrix}\mathcal{D}_1 & \mathcal{D}_2 \\ A_1 & A_2\end{matrix}}{A_1 \wedge A_2}\,\wedge I \qquad \dfrac{\begin{matrix}\mathcal{D} \\ A_1 \wedge A_2\end{matrix}}{A_1}\,\wedge E_1 \qquad \dfrac{\begin{matrix}\mathcal{D} \\ A_1 \wedge A_2\end{matrix}}{A_2}\,\wedge E_2$$

Implication rules

Implication rules

$$\frac{\begin{array}{c}[A]^m\\ \mathcal{D}\\ B\end{array}}{A \rightarrow B} \rightarrow I \qquad\qquad \frac{\begin{array}{c}\mathcal{D}_1\\ A \rightarrow B\end{array} \qquad A}{B} \rightarrow E$$

Disjunction rules

Disjunction rules

$$\dfrac{\begin{matrix}\mathcal{D}\\ A_1\end{matrix}}{A_1 \vee A_2}\ \vee I_1 \qquad \dfrac{\begin{matrix}\mathcal{D}\\ A_2\end{matrix}}{A_1 \vee A_2}\ \vee I_2 \qquad \dfrac{\begin{matrix}\\ \mathcal{D}_1\\ A_1 \vee A_2\end{matrix} \quad \begin{matrix}[A_1]^m\\ \mathcal{D}_2\\ B\end{matrix} \quad \begin{matrix}[A_2]^n\\ \mathcal{D}_3\\ B\end{matrix}}{B}\ \vee E\text{,m,n}$$

Alternative: $$\dfrac{\begin{matrix}\mathcal{D}_1\\ A_1 \vee A_2\end{matrix} \quad \begin{matrix}\mathcal{D}_2\\ A_1 \rightarrow B\end{matrix} \quad \begin{matrix}\mathcal{D}_3\\ A_2 \rightarrow B\end{matrix}}{B}\ \vee E$$

Super Theorem of Awesome

Theorem

Natural deduction is sound and complete with respect to propositional logic!!!!11one

Example 1

Example 1

Prove

$$\{a, a \rightarrow b\} \vdash a \wedge b$$

Example 1

Example 1

Prove

$$\{a, a \rightarrow b\} \vdash a \wedge b$$

$$\dfrac{a^{\overline{1}} \qquad \dfrac{a^{\overline{1}} \qquad a \rightarrow b^{\overline{2}}}{b} \rightarrow E}{a \wedge b} \wedge I$$

Example 2

Example 2

Prove

$$\emptyset \vdash (a \wedge b) \rightarrow b$$

Example 2

Example 2

Prove

$$\emptyset \vdash (a \land b) \to b$$

$$\cfrac{\cfrac{a \land b^{1}}{b}\land E_1}{(a \land b) \to b}\to I, 1$$

Example 3

Example 3

Prove

$$(a \wedge b) \vee (a \wedge c) \vdash c$$

Example 3

Example 3

Prove

$$(a \wedge b) \vee (a \wedge c) \vdash c$$

$$\dfrac{(a \wedge b) \vee (a \wedge c)^1 \qquad \dfrac{a \wedge b^2}{a}E_1{-}\wedge \qquad \dfrac{a \wedge c^3}{a}E_1{-}\wedge}{a} \vee E, 2, 3$$

Example 4

Example 4

Prove

$$\vdash (a \rightarrow b) \rightarrow (\neg b \rightarrow \neg a)$$

Example 4

Example 4

Prove

$$\vdash (a \to b) \to (\neg b \to \neg a)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{a^3 \qquad a \to b^1}{b} \to E \qquad \neg b \equiv b \to \bot^2}{\bot} \to E}{\neg a \equiv a \to \bot} \to I, 3}{(\neg b \to \neg a)} \to I, 2}{(a \to b) \to (\neg b \to \neg a)} \to I, 1$$

Last exercises, now with candy!

Exercise

1. $\vdash a \rightarrow (a \vee b)$
2. $\vdash (a \vee a) \rightarrow a$
3. $\vdash a \rightarrow (b \rightarrow a)$
4. $\vdash \neg(a \vee b) \rightarrow \neg a$
5. $\vdash \neg a \rightarrow (a \rightarrow b)$
6. $\vdash (b \rightarrow c) \rightarrow ((a \wedge b) \rightarrow c)$
7. $\vdash (a \rightarrow b) \rightarrow (a \rightarrow (b \vee c))$
8. $\vdash ((a \rightarrow b) \wedge (b \rightarrow c)) \rightarrow (a \rightarrow c)$
9. $\vdash ((a \rightarrow b) \wedge \neg b) \rightarrow \neg a$
10. $\vdash (a \rightarrow (b \rightarrow c)) \rightarrow ((a \rightarrow b) \rightarrow (a \rightarrow c))$
11. $\vdash a \leftrightarrow \neg\neg a$
12. $\vdash ((a \rightarrow b)) \leftrightarrow (\neg b \rightarrow \neg a)$
13. $\vdash a \vee \neg a$

Proof of correctness

All rules are sound

If $H_{d_i} \models \text{conc}(d_i)$ then $H_d \models \text{conc}(d)$.

Proof (sketch) of correctness

Rule for $\rightarrow E$.

- We have $\text{conc}(d) = B$, $\text{conc}(d_1) = A$, $\text{conc}(d_2) = A \rightarrow B$, and can assume $H_{d_1} \models A$ and $H_{d_2} \models A \rightarrow B$. Want to prove: $H_d \models A \rightarrow B$.
- We can see that $H_d = H_{d_1} \cup H_{d_2}$. By hypothesis, let V be s.t. $V \Vdash H_d$ (because w.t.p. $H_d \models ?$).
- Then, necessarily, $V \Vdash H_{d_1}$ and $V \Vdash H_{d_2}$, so that we know $V \Vdash A$ and $V \Vdash A \rightarrow B$. Therefore, $V \Vdash B$.
- Because we assumed $V \Vdash H_d$ and got $V \Vdash B$, we have $H_d \models B$.

Proof of correctness (continued)

Proof (sketch) of correctness, continued

Rule for $\rightarrow I$.

- We have $\text{conc}(d) = A \rightarrow B$, $\text{conc}(d_1) = B$ and can assume $H_{d_1} \models B$. Want to prove: $H_d \models A \rightarrow B$.
- We can see that $H_{d_1} \subset H_d \cup \{A\}$. By hypothesis, let V be s.t. $V \Vdash H_d$ (because w.t.p. $H_d \models ?$).
- Suppose $V \Vdash A$. Then, $V \Vdash H_{d_1}$ $(H_d \cup \{A\})$. In that case, we conclude that $V \models B$. Therefore, by definition of satisfaction, $V \Vdash A \rightarrow B$.
- Because we assumed $V \Vdash H_d$ and got $V \Vdash A \rightarrow B$, we have $H_d \models A \rightarrow B$.

Proof of correctness (continued)

Proof (sketch) of correctness, continued

Rule for $\bot$.

- We have $\mathrm{conc}(d) = A$, $\mathrm{conc}(d_1) = \bot$ and can assume $H_{d_1} \models \bot$. Want to prove: $H_d \models A$.
- We can see that $H_{d_1} \subset H_d \cup \{\neg A\}$. By hypothesis, let V be s.t. $V \Vdash H_d$ (because w.t.p. $H_d \models ?$).
- Let's assume $V \Vdash \neg A$. Then, $V \Vdash H_{d_1}$ $(H_d \cup \{\neg A\})$. Then, $V \Vdash \bot$. This is a contradiction. Therefore, $V \nVdash \neg A$, or $V \Vdash A$.
- Because we assumed $V \Vdash H_d$ and got $V \Vdash A$, we have $H_d \models A$.

What do we know so far?

- How do specify a logical language using syntax
- How to give it the intended meaning using semantics
- Truth tables
- Resolution for formulae in CNF
- Natural deduction as a *sound* and complete proof system

We have techniques to verify systems. What are we missing?

What do we know so far?

- How do specify a logical language using syntax
- How to give it the intended meaning using semantics
- Truth tables
- Resolution for formulae in CNF
- Natural deduction as a *sound* and complete proof system

We have techniques to verify systems. What are we missing?

Expressiveness!

Summary

1. Introduction
2. Syntax
3. Semantics
4. Verification
   - Truth Tables
   - Resolution
   - Natural Deduction
5. Beyond Propositional Logic
   - First Order Logic
   - Modal Logics
   - Dynamic Logic
   - Hybrid systems and Differential Dynamic Logic

Summary

1. Introduction
2. Syntax
3. Semantics
4. Verification
   - Truth Tables
   - Resolution
   - Natural Deduction
5. Beyond Propositional Logic
   - First Order Logic
   - Modal Logics
   - Dynamic Logic
   - Hybrid systems and Differential Dynamic Logic

First order logic

What if these weren't propositions? What if we could write them?

- $3^2 = 9$
- $\forall n \in \mathbb{N}_0, n \geq 0$
- $\forall x, y \in \mathbb{N}(x^2 + y^2 = z^2)$
- Any student is younger than any professor.

Core ideas of FOL

You add variables, $x \in X$!

- You keep the connectives
- You add properties: $p(x)$, $ismother(Anne, John)$
- You add functions: $s(x) = x + 1$, or $mother(John) = Anne$.
- You add quantifiers over variables: $\exists x A$, $\forall y B$

Variables $\neq$ propositions:

- Evaluate$(x) = v$, v is a value.
- Evaluate$(p) \in \{0, 1\}$

Examples

- John is a child, $C(John)$
- Anne is John's mother, $M(Anne, John)$
- Any child is younger than their mothers:

$$\forall x \forall y (C(x) \wedge M(y, x)) \rightarrow N(x, y)$$

- The function f is surjective:

$$\forall y \exists x f(x) = y$$

- The set has at least three different elements:

$$\exists x \exists y \exists z (\neg(x = y) \wedge \neg(x = z) \wedge \neg(y = z))$$

Exercises (but it's almost over anyways)

Exercises

1. A and B are sons of C
2. Since noone is its own ancestor, if A is an ancestor of B, then B isn't an ancestor of A
3. Sons of the same mother are brothers
4. No even number is a prime
5. Not all primes are odd
6. Any prime is equal to 2, or odd
7. Any transitive, anti-reflexive (binary) relation is anti-symmetric
8. Every hour someone is robbed. We'll meet him today

Interpreting terms

We need the following new elements:

- Variables take values in a given domain/universe U
- To keep track of variables, we need $\rho : X \to U$
- Instead of valuation V, we have an interpretation I that also handles $p(x, y)$ and $f(x, y)$

Definition

Let $\mathcal{M} = (U, I)$ be an *interpretation structure*. Interpreting terms is defined as follows:

- $[\![x]\!]^{\rho}_{\mathcal{M}} = \rho(x)$, for $x \in X$
- $[\![c]\!]^{\rho}_{\mathcal{M}} = I(c)$, for a constant c (what *is* a constant?)
- $[\![f(t_1, ..., t_n)]\!]^{\rho}_{\mathcal{M}} = I(f)([\![t_1]\!]^{\rho}_{\mathcal{M}}, ..., [\![t_n]\!]^{\rho}_{\mathcal{M}})$, for a function f of arity n

Example

- $I(\bar{i}) = i \in U = \mathbb{N}$, from a symbol to a number (these are constants)
- $\rho(x) = 3$ and $\rho(y) = 1$
- $I(\oplus)(w, z) = w + z$ (from the *symbol* $\oplus$ to the *meaning* of $+$!)

Example

Let's interpret $x \oplus (\bar{2} \oplus y)$. (why can't I write 2 instead?)

Example

- $I(\bar{i}) = i \in U = \mathbb{N}$, from a symbol to a number (these are constants)
- $\rho(x) = 3$ and $\rho(y) = 1$
- $I(\oplus)(w, z) = w + z$ (from the *symbol* $\oplus$ to the *meaning* of $+$!)

Example

Let's interpret $x \oplus (\overline{2} \oplus y)$. (why can't I write 2 instead?)

$$\begin{aligned}
\llbracket x \oplus (\overline{2} \oplus y) \rrbracket^{\rho}_{(U,I)} &= I(\oplus)(\llbracket x \rrbracket^{\rho}_{(U,I)}, \llbracket \overline{1} \oplus y \rrbracket^{\rho}_{(U,I)}) \\
&= \llbracket x \rrbracket^{\rho}_{(U,I)} + \llbracket \overline{2} \oplus y \rrbracket^{\rho}_{(U,I)} \\
&= \rho(x) + I(\oplus)(\llbracket \overline{2} \rrbracket^{\rho}_{(U,I)}, \llbracket y \rrbracket^{\rho}_{(U,I)}) \\
&= 3 + \llbracket \overline{2} \rrbracket^{\rho}_{(U,I)} + \llbracket y \rrbracket^{\rho}_{(U,I)} \\
&= 3 + I(\overline{2}) + \rho(y) \\
&= 3 + 2 + 1 = 6
\end{aligned}$$

Satisfying formulae

Definition (satisfaction)

Let $\mathcal{M} = (U, I)$. Besides the rules from propositional logic:

- $\mathcal{M}, \rho \Vdash P(t_1, ..., t_n)$ if $I(P)([\![t_1]\!]^{\rho}_{\mathcal{M}}, ..., [\![t_n]\!]^{\rho}_{\mathcal{M}}) = 1$
- $\mathcal{M}, \rho \Vdash \forall x\, A$ if for all $u \in U$, $\mathcal{M}, \rho[x \mapsto u] \Vdash A$
- $\mathcal{M}, \rho \Vdash \exists x\, A$ if for some $u \in U$, $\mathcal{M}, \rho[x \mapsto u] \Vdash A$

Very quick exercises

Convert the following formulae into equivalents with the other quantifier:

1. $\neg \forall x A$
2. $\exists x A$

Satisfying formulae

Definition (satisfaction)

Let $\mathcal{M} = (U, I)$. Besides the rules from propositional logic:

- $\mathcal{M}, \rho \Vdash P(t_1, ..., t_n)$ if $I(P)(\llbracket t_1 \rrbracket^{\rho}_{\mathcal{M}}, ..., \llbracket t_n \rrbracket^{\rho}_{\mathcal{M}}) = 1$
- $\mathcal{M}, \rho \Vdash \forall x\, A$ if for all $u \in U$, $\mathcal{M}, \rho[x \mapsto u] \Vdash A$
- $\mathcal{M}, \rho \Vdash \exists x\, A$ if for some $u \in U$, $\mathcal{M}, \rho[x \mapsto u] \Vdash A$

Very quick exercises

Convert the following formulae into equivalents with the other quantifier:

1. $\neg\forall x A$
2. $\exists x A$

Answers: $\exists x \neg A$ and $\neg\forall x \neg A$

Final words on FOL

- A LOT more expressive
- There are proof systems that are sound and complete!
- Semidecidable:
  - If A is a theorem, you can find a proof
  - If A is not a theorem, the algorithm may not answer

Most complex logics become undecidable :(

Summary

1. Introduction
2. Syntax
3. Semantics
4. Verification
   - Truth Tables
   - Resolution
   - Natural Deduction
5. Beyond Propositional Logic
   - First Order Logic
   - Modal Logics
   - Dynamic Logic
   - Hybrid systems and Differential Dynamic Logic

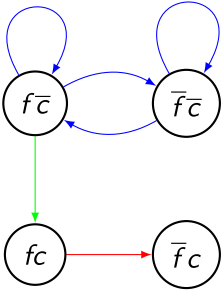
Modal (Propositional) Logic: core ideas

What if instead of one world, we had several “possible worlds”?

- Cars aren’t always going fast f
- Cars don’t always crash, c
- Perhaps $V \Vdash f$ isn’t always the case...
- What if we represent each V explicitly?
- What if we can talk about them *within* the logic itself!?

Valuception... *cunning!*

Example: cars crashing



- If the car is fast and crashed, it will probably skid to a stop
- If a car is going fast, it may crash
- A car may brake, accelerate, or keep its speed

Notice the car won't crash if it is going slow!

How do we talk about these different worlds?

We use modalities:

- $\Box A$ means that A is *necessary*
- $\Diamond A$ means that A is *possible*

How do you think the semantics work?

How do we talk about these different worlds?

We use modalities:

- $\Box A$ means that A is *necessary*
- $\Diamond A$ means that A is *possible*

How do you think the semantics work?

Semantics

Let $F = \langle G, R, \models \rangle$ be a frame. G is the set of possible worlds. R is the accessibility relation. $w \models p$, with $w \in G$ means that p is true in w.

- $w \models \Box A$ if whenever $(w, v) \in R$ then $v \models A$.
- $w \models \Diamond A$ if there is some $(w, v) \in R$ such that $v \models A$.

Plus the usual propositional logic.

Restrictions on the frame

If the frame is arbitrary, we have the following properties:

- If $\vdash A$, then $\vdash \Box A$ (Necessitation rule)
- $\Box(A \vee B) \to (\Box A \to \Box B)$ (Distribution Axiom)

The more restrictions you put in your frame, the more axioms you get:

- If R is reflexive, then $\Box A \to A$
- You can get up to making R an equivalence relation, and get $\Box A \to \Box\Box A$, $\Diamond A \to \Box\Diamond P$.

Final thoughts

- Modalities can have many meanings:
    - Knowledge
    - Belief
    - Necessity/possibility
    - Temporal
    - Etc...
- But suppose we wanted to change our location?
- Or suppose that we want to specify how R is defined?
- What if R was dynamic?

Summary

1. Introduction
2. Syntax
3. Semantics
4. Verification
    - Truth Tables
    - Resolution
    - Natural Deduction
5. Beyond Propositional Logic
    - First Order Logic
    - Modal Logics
    - Dynamic Logic
    - Hybrid systems and Differential Dynamic Logic

Dynamic Logic: core ideas

- Instead of propositions, we've got variables
- Inside the $\Box$, we put *programs*!!!
- The programs tells us what R should be!

Man, if this is not exciting, I don't know WHAT is! ☺

Programs

- D.L. was defined to be able to reason about computer programs.
- Therefore, the programs we will use are similar to computer programs

Programs

Basic actions:

- Assignment: $x := e$, where x is a variable and e is an expression made from other variables and the usual operators (+, -, etc).
- Test: $?cond$, where $cond$ is some condition, such as $x = 3$, or $x < 0$.
- NOP: $\mathbf{1}$, does nothing
- BLOCK: $\mathbf{0}$, an action that results in contradiction

And compound actions:

- Sequence: $a; b$, means b executes after a
- Choice: $a \cup b$, the program can perform either of the two actions
- Iteration: $a*$ runs a zero or more times sequentially.

A simple example

Cars have speed and position, p and v. The wind might affect the car.

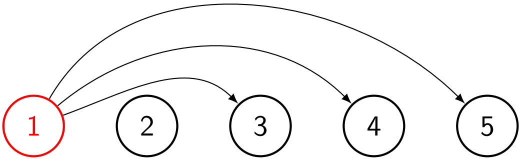
- When time passes, the car moves, but might be affected by the wind:

$$(p := p + v) \cup (p := p - 1) \cup (p := p + 1)$$

- The driver may also decide to accelerate or brake:

$$(v := v + 1) \cup (v := v - 1)$$

Suppose v is 2. Numbers represent p. Here's $(p := p + v) \cup (p := p - 1) \cup (p := p + 1)$:



A simple example

Cars have speed and position, p and v. The wind might affect the car.

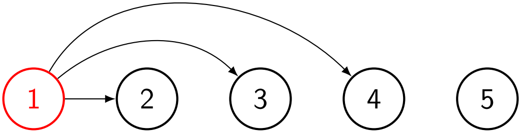
- When time passes, the car moves, but might be affected by the wind:

$$(p := p + v) \cup (p := p - 1) \cup (p := p + 1)$$

- The driver may also decide to accelerate or brake:

$$(v := v + 1) \cup (v := v - 1)$$

Suppose v is 2. Numbers represent p. Here's
$v := v - 1; ((p := p + v) \cup (p := p - 1) \cup (p := p + 1))$:



Some interesting axioms

The following axioms might help understand how programs interact with modalities:

- $[0]A$
- $[1]A \equiv A$
- $[a \cup b]A \equiv [a]A \wedge [b]A$
- $[a; b]A \equiv [a]([b]A)$
- $[a*]A \equiv A \wedge [a][a*]A$
- $A \wedge [a*](A \rightarrow [a]A) \rightarrow [a*]A$ (what does this look like?)

Quick exercise

Define the program that represents:

if A then a else b

What do we have so far?

- You know how to start from propositional logic
- ... then build in modalities
- ... then build in even more complex and dynamic modalities
- All these logics have axiomatisations/proof systems
- They can also be extended to first-order variations

Can we accurately model a car and car cruise control with what we have?

Continuous time

NO

Physics happen in continuous time.

Summary

1. Introduction
2. Syntax
3. Semantics
4. Verification
   - Truth Tables
   - Resolution
   - Natural Deduction
5. Beyond Propositional Logic
   - First Order Logic
   - Modal Logics
   - Dynamic Logic
   - Hybrid systems and Differential Dynamic Logic

Cyberphysical Systems and Hybrid Systems

What are cyberphysical systems?

- They are real-world systems that have behaviour that occurs in continuous time (i.e. in $\mathbb{R}$)
- .. but they also have behaviours that occur in no time at all, such as computation (i.e. how long does $x := e$ take? None!)

It is imperative that we deal with cars moving in continuous time, or they may crash between time-steps.

We call the models that we use to represent cyber-physical systems *hybrid systems*

Differential Dynamic Logic: core ideas

- Extends programs with a notion of continuous time
- Its programs become *hybrid*: they feature both continuous and discrete dynamics

Add to the programs the following operation:

$$(x' = \theta \,\&\, \chi)$$

Differential equations specify how each variable evolves over time, and allows time to pass until χ ceases to hold.

Example: cars moving, avoid crashing

Two cars, each with position x_i, speed v_i and acceleration a_i:

$$(x_i' = v, v_i' = a_i \,\&\, x_1 < x_2)$$

Discussion and results

- There is a proof system that is sound and “relatively complete”. There is a program to (almost) automatically verify formulae
- Still very hard to do
- Very recent research

- Examples of hybrid systems?
- Examples of how to extend these logics?

It's been wonderful being here!
Thank you! ☺

Slides at: `http://www.cs.cmu.edu/~jmartins/ideamath/slides.pdf`

If you ever have any questions about logic, e-mail me:
jmartins@cs.cmu.edu