

COSC 221 - Introduction to Discrete Structures

Lecture - Logic-04

Readings

- ▶ Propositional Logic: Sections 3.1, 3.2, 3.3
- ▶ Predicate Logic: Sections 3.4
- Computer Science Connections
 - 1. Computational Complexity (Section 3.3)
 - 2. Modern Compilers (Section 3.3)
- 3. Game Trees (Section 3.4)



Arguments and Logic Inference

- ▶ Argument Form
- ▶ Inference Rules

Not covered in detail in the textbook, but important!



Arguments and Logic Inference

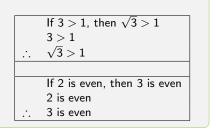
- ▶ Argument Form
- ▶ Inference Rules

Not covered in detail in the textbook, but important!

If 3 > 1, then $\sqrt{3} > 1$ 3 > 1 $\therefore \sqrt{3} > 1$ If 2 is even, then 3 is even 2 is even \therefore 3 is even



- ▶ Argument Form
- ▶ Inference Rules



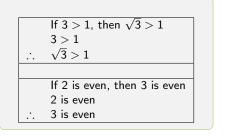
Arguments and Logic Inference

Same pattern, different contexts

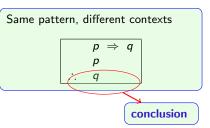
$$\begin{array}{c} p \Rightarrow q \\ p \\ \therefore q \end{array}$$



- ▶ Argument Form
- ▶ Inference Rules



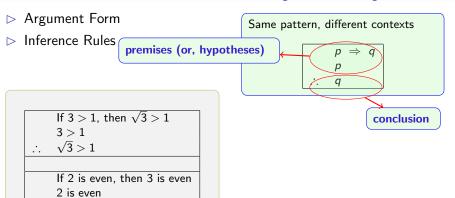
Arguments and Logic Inference



3 is even



Arguments and Logic Inference





Arguments and Logic Inference

Argument Form

Same pattern, different contexts

> Inference Rules

premises (or, hypotheses)

 $p \Rightarrow \hat{q}$ p q

conclusion

Argument Form: $\varphi_1, \ \varphi_2, \ \cdots, \ \varphi_{n-1}, \ \varphi_n$



Arguments and Logic Inference

Argument Form

Same pattern, different contexts

> Inference Rules

premises (or, hypotheses)

 $p \Rightarrow \hat{q}$ p q

conclusion

Argument Form: $\varphi_1, \ \varphi_2, \ \cdots, \ \varphi_{n-1}, \ \varphi_n$



Arguments and Logic Inference

Argument Form

Same pattern, different contexts

> Inference Rules

 $\begin{array}{c} \text{premises (or, hypotheses)} \\ \hline p \Rightarrow q \\ \hline p \\ \hline q \end{array}$

conclusion

Valid Argument Form:

Whenever the premises are all true, the conclusion is true.

Argument Form: $\varphi_1, \ \varphi_2, \ \cdots, \ \varphi_{n-1}, \ \varphi_n$



- ▶ Inference Rules

- Q.1) Is this argument form valid?
 - A) Yes
 - B) No
 - C) You haven't taught us yet!
 - D) Whatever!

Arguments and Logic Inference

Same pattern, different contexts

$$\begin{array}{c|c}
p \Rightarrow q \\
p \\
\vdots & q
\end{array}$$

Valid Argument Form:

Whenever the premises are all true, the conclusion is true.

Argument Form: $\varphi_1, \varphi_2, \cdots, \varphi_{n-1}, \varphi_n$



- > Argument Form
- ▶ Inference Rules

р	q	$p \Rightarrow q$	р	q
Т	Т	Т	Т	Т
Т	F	F	Т	F
F	Т	Т	F	Т
F	F	Т	F	F

Arguments and Logic Inference

Same pattern, different contexts

$$\begin{array}{c} p \Rightarrow q \\ p \\ \therefore q \end{array}$$

Valid Argument Form:

Whenever the premises are all true, the conclusion is true.

- ▶ Truth-Table



Arguments and Logic Inference

Argument Form

Same pattern, different contexts

Inference Rules

premises (or, hypotheses)	premises	(or,	hypotheses)	
---------------------------	----------	------	-------------	--

р	q	$p \Rightarrow q$	р	q
Т	Т	Т	Т	Т
Т	F	F	Т	F
F	Т	Т	F	Т

р	q	$p \Rightarrow q$	р	q
Т	Т	Т	Т	Т
Т	F	F	Т	F
F	Т	Т	F	Т
F	F	T	F	F

- ▶ Truth-Table
- ▶ Tautology



Arguments and Logic Inference

- Argument Form
- Inference Rules

>	intere	ence	Rules	prer	nises	(or, h	ypothe	eses)
	p	q	$p \Rightarrow$	q	р	q		
	Т	т	Т		Т	Т		

Т	Т	Т	Т	Т
Т	F	F	Т	F
F	Т	Т	F	Т

Same pattern, o	different	contexts
-----------------	-----------	----------

Conclusion

- ▶ Truth-Table
- ▶ Tautology



Conclusion

Arguments and Logic Inference

> Argument Form

 ${\sf Same \ pattern, \ different \ contexts}$

▶ Inference Rules

premises (or, hypotheses)

 $\Rightarrow q \mid p \mid q$

T F F T F

Critical Row: premises are all true

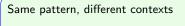
- ▶ Truth-Table
- ▶ Tautology



- > Argument Form
- ▶ Inference Rules

р	q	$p \Rightarrow q$	р	q
T	Т	Т	Т	Т
T	F	F	Т	F
F	Т	Т	F	Т
F	F	Т	F	F

Arguments and Logic Inference





Critical Row: premises are all true

Testing Validity of Argument Forms

- ▶ Truth-Table
- ▶ Tautology

An argument form is valid iff the conclusion is true in every **critical row**

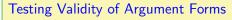


- ▶ Inference Rules

Arguments and Logic Inference

Same pattern, different contexts

$$\begin{array}{c}
p \Rightarrow q \\
p \\
\therefore q
\end{array}$$



- ▶ Truth-Table
- Tautology

An argument form is valid iff it is a tautology when treated as a conditional proposition.



- > Argument Form
- ▶ Inference Rules

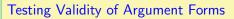
For example,

$$(p \Rightarrow q) \land p \Rightarrow q$$

is indeed a tautology.

Arguments and Logic Inference

Same pattern, different contexts



- ▶ Truth-Table
- Tautology

An argument form is valid iff it is a tautology when treated as a conditional proposition.

- Modus Ponens
- Modus Tollens
- Elimination
- Transitivity

Inference Rules

Inference Rule: Valid Argument Form



- ▶ Modus Tollens
- ▶ Elimination
- ▶ Transitivity

Inference Rule: Valid Argument Form

Modus Ponens

(Latin meaning: method of affirming)

Inference Rules

- ▶ Modus Tollens
- ▶ Elimination
- ▶ Transitivity

Inference Rule: Valid Argument Form

Sound Argument

- ∇alid argument form
- Premises are all true

Modus Ponens

From Argument Form to Argument:

Substitute propositions for proposition variables

$$\begin{array}{c|c} p \Rightarrow q & \text{If } 3 > 1 \text{, then } \sqrt{3} > 1 \\ p & 3 > 1 \\ q & \ddots & \sqrt{3} > 1 \end{array}$$



- Modus Ponens
- ▶ Elimination
- ▶ Transitivity

Inference Rule: Valid Argument Form

Sound Argument

- ∇alid argument form
- Premises are all true

Modus Tollens

(Latin meaning: method of denying)

$$egin{array}{c|c|c|c} p \Rightarrow q & & ext{If } \sqrt{.5} > 1, ext{ then } .5 > 1 \\ \neg q & & ext{.5} \leq 1 \\ \therefore & \neg p & & \therefore & \sqrt{.5} \leq 1 \end{array}$$



- Modus Ponens
- Modus Tollens
- → Elimination
- ▶ Transitivity

Inference Rule: Valid Argument Form

Sound Argument

- ∇alid argument form
- ▶ Premises are all true

Elimination

$$\begin{array}{c|cccc} p \lor q & & (x > 1) \lor (y > 1) \\ \neg q & & x \le 1 \\ \therefore & p & & \therefore & y > 1 \end{array}$$



- Modus Ponens
- Modus Tollens
- ▶ Elimination

Inference Rules

Inference Rule: Valid Argument Form

Sound Argument

- ∇alid argument form
- ▶ Premises are all true

Transitivity

	$p \Rightarrow q$		If $x > 1$, then $ x > 1$
	$q \Rightarrow r$		If $ x > 1$, then $x \neq 0$
·:.	$p \Rightarrow r$	∴	If $x > 1$, then $x \neq 0$

Additional Notes Other Useful Inference Rules

Generalization

Specialization

$$\begin{array}{c|cccc} p & \wedge & q & & (x > 1) & \wedge & x \text{ even} \\ \therefore & p & & \ddots & x > 1 & \end{array}$$

Additional Notes Other Useful Inference Rules

Proof by Division into Cases

$$\begin{array}{|c|c|c|c|c|} \hline p \lor q \\ p \Rightarrow r \\ q \Rightarrow r \\ \therefore r \\ \hline \end{array} \begin{array}{|c|c|c|c|} \hline (x \ge 1) \lor (x \le -1) \\ \text{If } x \ge 1 \text{, then } x^2 \ge 1 \\ \text{If } x \le -1 \text{, then } x^2 \ge 1 \\ \hline \therefore x^2 \ge 1 \\ \hline \end{array}$$

Exercise: Use the truth-table method to show the above is valid.

Contradiction Rule

$$\begin{array}{ccc}
\neg p \Rightarrow F \\
\therefore p
\end{array}$$

UBC

Use Inference Rules to Construct Complex Arguments

The Story: Finding my glasses?

RK = "I was reading in the kitchen"

RL = "I was reading in the living room"

 $\mathsf{GK} = \mathsf{"My} \; \mathsf{glasses} \; \mathsf{are} \; \mathsf{on} \; \mathsf{the} \; \mathsf{kitchen} \; \mathsf{table"}$

 $\mathsf{GC} = \mathsf{``My} \mathsf{\ glasses} \mathsf{\ are} \mathsf{\ on \ the \ coffee \ table''}$

SB = "I saw them at breakfast"



Use Inference Rules to Construct Complex Arguments

The Story: Finding my glasses?

RK = "I was reading in the kitchen"

 $\mathsf{RL} = \mathsf{"I}$ was reading in the living room"

 $\mathsf{GK} = \mathsf{"My} \; \mathsf{glasses} \; \mathsf{are} \; \mathsf{on} \; \mathsf{the} \; \mathsf{kitchen} \; \mathsf{table"}$

 $\mathsf{GC} = \mathsf{``My} \mathsf{ glasses} \mathsf{ are} \mathsf{ on} \mathsf{ the} \mathsf{ coffee} \mathsf{ table''}$

SB = "I saw them at breakfast"

Known Facts

- $ightharpoonup RK \Rightarrow GK$
- $ightharpoonup GK \Rightarrow SB$
- ¬SB
- ► RK ∨ RL
- $ightharpoonup RL \Rightarrow GC$



Use Inference Rules to Construct Complex Arguments

The Story: Finding my glasses?

RK = "I was reading in the kitchen"

RL = "I was reading in the living room"

 $\mathsf{GK} = \mathsf{"My} \; \mathsf{glasses} \; \mathsf{are} \; \mathsf{on} \; \mathsf{the} \; \mathsf{kitchen} \; \mathsf{table"}$

 $\mathsf{GC} = \mathsf{``My} \mathsf{\ glasses} \mathsf{\ are} \mathsf{\ on} \mathsf{\ the} \mathsf{\ coffee} \mathsf{\ table''}$

SB = "I saw them at breakfast"

Known Facts

- $ightharpoonup RK \Rightarrow GK$
- $ightharpoonup GK \Rightarrow SB$
- ¬SB
- ► RK ∨ RL
- $ightharpoonup RL \Rightarrow GC$

- Q.2) Inference rule used is
 - A) transitivity
 - B) modus tollens
 - C) elimination
 - D) modus ponens

Step 1

$$RK \Rightarrow GK$$

 $GK \Rightarrow SB$

 \therefore RK \Rightarrow SB



Use Inference Rules to Construct Complex Arguments

The Story: Finding my glasses?

RK = "I was reading in the kitchen"

RL = "I was reading in the living room"

 $\mathsf{GK} = \mathsf{"My} \; \mathsf{glasses} \; \mathsf{are} \; \mathsf{on} \; \mathsf{the} \; \mathsf{kitchen} \; \mathsf{table"}$

 $\mathsf{GC} = \mathsf{``My} \mathsf{\ glasses} \mathsf{\ are} \mathsf{\ on} \mathsf{\ the} \mathsf{\ coffee} \mathsf{\ table''}$

SB = "I saw them at breakfast"

Known Facts

- $ightharpoonup RK \Rightarrow GK$
- $ightharpoonup GK \Rightarrow SB$
- ¬SB
- ► RK ∨ RL
- $ightharpoonup RL \Rightarrow GC$

- Q.3) Inference rule used is
 - A) transitivity
 - B) modus tollens
 - C) elimination
 - D) modus ponens

Step 2

$$RK \Rightarrow SB$$

 $\neg SB$



Use Inference Rules to Construct Complex Arguments

The Story: Finding my glasses?

RK = "I was reading in the kitchen"

RL = "I was reading in the living room"

 $\mathsf{GK} = \mathsf{"My} \; \mathsf{glasses} \; \mathsf{are} \; \mathsf{on} \; \mathsf{the} \; \mathsf{kitchen} \; \mathsf{table"}$

 $\mathsf{GC} = \mathsf{``My} \mathsf{\ glasses} \mathsf{\ are} \mathsf{\ on} \mathsf{\ the} \mathsf{\ coffee} \mathsf{\ table''}$

SB = "I saw them at breakfast"

Known Facts

- $ightharpoonup RK \Rightarrow GK$
- $ightharpoonup GK \Rightarrow SB$
- ¬SB
- ► RK ∨ RL
- $ightharpoonup RL \Rightarrow GC$

- Q.4) Inference rule used is
 - A) transitivity
 - B) modus tollens
 - C) elimination
 - D) modus ponens

Step 3

¬RK RL ∨ RK RI



Use Inference Rules to Construct Complex Arguments

The Story: Finding my glasses?

RK = "I was reading in the kitchen"

RL = "I was reading in the living room"

 $\mathsf{GK} = \mathsf{"My} \; \mathsf{glasses} \; \mathsf{are} \; \mathsf{on} \; \mathsf{the} \; \mathsf{kitchen} \; \mathsf{table"}$

 $\mathsf{GC} = \mathsf{``My} \mathsf{\ glasses} \mathsf{\ are} \mathsf{\ on} \mathsf{\ the} \mathsf{\ coffee} \mathsf{\ table''}$

SB = "I saw them at breakfast"

Known Facts

- $ightharpoonup RK \Rightarrow GK$
- $ightharpoonup GK \Rightarrow SB$
- ¬SB
- ► RK ∨ RL
- $ightharpoonup RL \Rightarrow GC$

- Q.5) Inference rule used is
 - A) transitivity
 - B) modus tollens
 - C) elimination
 - D) modus ponens

Step 4

$$RL \Rightarrow GC$$

RL

. *GC*



Use Inference Rules to Construct Complex Arguments

The Story: Finding my glasses?

RK = "I was reading in the kitchen"

RL = "I was reading in the living room"

 $\mathsf{GK} = \mathsf{"My} \; \mathsf{glasses} \; \mathsf{are} \; \mathsf{on} \; \mathsf{the} \; \mathsf{kitchen} \; \mathsf{table"}$

 $\mathsf{GC} = \mathsf{``My} \mathsf{\ glasses} \mathsf{\ are} \mathsf{\ on} \mathsf{\ the} \mathsf{\ coffee} \mathsf{\ table''}$

SB = "I saw them at breakfast"

Known Facts

- $ightharpoonup RK \Rightarrow GK$
- $ightharpoonup GK \Rightarrow SB$
- ¬SB
- ► RK ∨ RL
- $ightharpoonup RL \Rightarrow GC$

Put T	ogether: the a	rgument
(1)	$RK \Rightarrow GK$	Given
(2)	$GK \Rightarrow SB$	Given
(3)	$RK \Rightarrow SB$	transitivity
(4)	$\neg SB$	Given
(5)	$\neg RK$	moduc tollens: (3), (4)
(6)	$RL \lor RK$	Given
(7)	RL	elimination: (5), (6)
(8)	$RL \Rightarrow GC$	Given
<u> </u>	GC	modus ponens:(7), (8)



Use Inference Rules to Construct Arguments

Let x, y, z be three particular integers, satisfying

- $\triangleright x < y < z$
- $\triangleright z < 1$
- |x| > 1

To show that x < -1.

True Propositions

- (a) $(x > 1) \Rightarrow (y > 1)$
- (b) $(y > 1) \Rightarrow (z > 1)$
- (c) $(z \le 1)$
- (d) $(x < -1) \lor (x > 1)$



Use Inference Rules to Construct Arguments

Let x, y, z be three particular integers, satisfying

$$\triangleright x < y < z$$

$$\triangleright z < 1$$

To show that x < -1.

True Propositions

(a)
$$(x > 1) \Rightarrow (y > 1)$$

(b)
$$(y > 1) \Rightarrow (z > 1)$$

(c)
$$(z \le 1)$$

(d)
$$(x < -1) \lor (x > 1)$$

The Argument

(1)
$$(x > 1) \Rightarrow (y > 1)$$
 | Given

(2)
$$(y > 1) \Rightarrow (z > 1)$$
 Given

(3)
$$(x > 1) \Rightarrow (z > 1)$$
 Transitivity

$$(3) \quad (x > 1) \Rightarrow (z > 1)$$

$$(4) \quad \neg(z>1)$$

$$(5) \neg (x > 1)$$

(6)
$$(x < -1) \lor (x > 1)$$
 Given

$$(0) \quad (x < -1) \quad \forall (x > 1)$$

$$\therefore \quad (x < -1)$$

$$\therefore$$
 $(x < -1)$

(4)
$$\neg (z > 1)$$
 Given
(5) $\neg (x > 1)$ Modus Tollens: (3),(4)

Elimination:
$$(5)$$
, (6)



Use Inference Rules to Construct Complex Arguments

Knights always tell truth and Knaves always tell lie. A says "B is a knight" B says "A and I are of opposite type".

What are A and B?



Use Inference Rules to Construct Complex Arguments

Knights always tell truth and Knaves always tell lie. A says "B is a knight" B says "A and I are of opposite type".

What are A and B?

► Suppose *A* is a knight. *B* is a knight and "A and I are of opposite types" is true. A contradiction!



Use Inference Rules to Construct Complex Arguments

Knights always tell truth and Knaves always tell lie. A says "B is a knight" B says "A and I are of opposite type".

What are A and B?

- ► Suppose *A* is a knight. *B* is a knight and "A and I are of opposite types" is true. A contradiction!
- ∴ A is NOT a knight



Use Inference Rules to Construct Complex Arguments

Knights always tell truth and Knaves always tell lie. A says "B is a knight" B says "A and I are of opposite type".

What are A and B?

- ► Suppose *A* is a knight. *B* is a knight and "A and I are of opposite types" is true. A contradiction!
- → ∴ A is NOT a knight

Q.6) The inference rule used is

- A) Modus Ponens
- B) Elimination
- C) Contradiction
- D) Generalization



Use Inference Rules to Construct Complex Arguments

Knights always tell truth and Knaves always tell lie. A says "B is a knight" B says "A and I are of opposite type".

What are A and B?

- Suppose A is a knight. B is a knight and "A and I are of opposite types" is true. A contradiction!
- → ∴ A is NOT a knight
- A is a knight or A is a knave



Use Inference Rules to Construct Complex Arguments

Knights always tell truth and Knaves always tell lie. A says "B is a knight" B says "A and I are of opposite type".

What are A and B?

- ► Suppose *A* is a knight. *B* is a knight and "A and I are of opposite types" is true. A contradiction!
- → ∴ A is NOT a knight
- A is a knight or A is a knave
- ∴ A is a knave



Use Inference Rules to Construct Complex Arguments

Knights always tell truth and Knaves always tell lie. A says "B is a knight" B says "A and I are of opposite type".

What are A and B?

- Suppose A is a knight. B is a knight and "A and I are of opposite types" is true. A contradiction!
- → ∴ A is NOT a knight
- ► A is a knight or A is a knave
- ∴ A is a knave

Q.7) The inference rule used is

- A) Modus Ponens
- B) Elimination
- C) Contradiction
- D) Generalization

Additional Notes Common Errors: Logical Fallacies

Converse Error	Inverse Error
$p \Rightarrow q$	$p \Rightarrow q$
q	$\neg p$
∴. p	.∵. ¬q
(affirming the consequence)	(denying the antecedent)



Summary and Expectations

Concepts

- Propositions and Connectives

Arguments and Logic Inference

- Argument and Argument Form
- ▶ Inference Rules

Modelling and Inferences

- iCliker exercises

- \triangleright Statements \Rightarrow Propositions
- Negation/Contrapositive of Compound (Conditional) Propositions
- Use of trueth tables and/or logic laws



Summary and Expectations

Concepts

- Propositions and Connectives
- > Truth Values and Truth Tables

Arguments and Logic Inference

- > Argument and Argument Form
- Inference Rules

Modelling and Inferences

- iCliker exercises

- Truth tables and Valid Argument Forms

Additional Notes

Announcement 1: Quiz 1 (Monday, Feb 5th)

- Propositional Logic
- Basics of Predicate Logic



Logic and Al Reasoning

Logics and Automated Reasoning in Al

Knowledge Representation: KB (Knowledge Base)

Query and Question

Algorithmic Problem

Inference Engine



Logic and Al Reasoning

Logics and Automated Reasoning in Al

Knowledge Representation: KB (Knowledge Base)

Query and Question Conjunction of facts and inference rules

Algorithmic Problem

Inference Engine



Logic and Al Reasoning

Logics and Automated Reasoning in Al

Knowledge Representation: KB (Knowledge Base)

Query and Question Given a statement Q, is $KB \Rightarrow Q$ true?

Algorithmic Problem

Inference Engine





Logics and Automated Reasoning in Al	
Knowledge Representation: KB (Knowledge Base)	
Query and Question	Given a statement Q, is $KB \Rightarrow Q$ true?
Algorithmic Problem	
Inference Engine	Is $KB \cap \neg Q$ False?
	Why does this work?



Logics and Automated Reasoning in Al

Knowledge Representation: KB (Knowledge Base)

Query and Question

Given a statement Q, is $KB \Rightarrow Q$ true?

Algorithmic Problem

Inference Engine

A solver for the satisfiability problem.

E.g.,
$$F = (x_1 \lor x_2) \land (x_8 \lor \neg x_{15} \lor x_{50}) \cdots (\neg x_{99} \lor x_{120})$$



Logic and Programming

- ► Imperative Programming
- Declarative Programming



Logic and Programming

- ► Imperative Programming
- Declarative Programming

Java, C, C#, etc. Instructions telling computers "how to do things"



Logic and Programming

- Imperative Programming
- Declarative Programming

Java, C, C#, etc.
Instructions telling computers "how to do things"

Prolog, LISP.
Statements telling computers "what to do"



Logic and Programming

- Imperative Programming
- Declarative Programming

Java, C, C#, etc.
Instructions telling computers "how to do things"

Prolog, LISP.

Statements telling computers "what to do"

Prolog - Programming in Logic

- Developed in 1970s
- Used in some areas of AI



Logic and Programming

- Imperative Programming
- ► Declarative Programming

```
% A Mini Knowledgebase about Java Classes
```

```
is-a(X, Y):- extends(X, Y).
is-a(X, Y):- extends(X, Z), is-a(Z, Y).
is-a(X, Y):- interface(Y), implements(X, Y).
```

```
%Facts
extends(mylist, arraylist).
extends(arraylist, abstractlist).
extends(error, throwable).
extends(exception, throwable).
```

%Rules

Prolog - Programming in Logic

- ► Developed in 1970s
- Used in some areas of AI

implements(abstractlist, list).
implements(throwable, serializable).
interface(list).



- ► Imperative Programming
- Declarative Programming

Propositions

E.g. "mylist extends arraylist"

```
% A Mini Knowledgebase about Java Classes
%Rules
is-a(X, Y):=extends(X, Y).
is-a(X, Y):= extends(X, Z), is-a(Z, Y).
is-a(X, Y):- interface(Y), implements(X, Y).
%Facts
extends(mylist, arraylist).
extends(arraylist, abstractlist).
extends (error, throwalle).
extends (exception, throwable).
implements(abstractlist, list).
implements (throwable / serializable).
interface(list)
```

Logic and Programming



- Imperative Programming
- Declarative Programming

Propositions?

E.g. "X is a Y if X extends Y"

Remember "He is a student"

Propositions

E.g. "mylist extends arraylist"

Logic and Programming

% A Mini Knowledgebase about Java Classes

%Rules

```
is-a(X, Y):- extends(X, Y).
is-a(X, Y):- extends(X, Z), is-a(Z, Y).
is-a(X, Y):- interface(Y), implements(X, Y).
```

%Facts

```
extends(mylist, arraylist).
extends(arraylist, abstractlist).
extends(error, throwable).
extends(exception, throwable).
```

```
implements(abstractlist, list).
implements(throwable serializable).
interface(list).
```



Logic and Programming

- Imperative Programming
- ► Declarative Programming

```
% A Mini Knowledgebase about Java Classes
%Rules
is-a(X, Y):- extends(X, Y).
is-a(X, Y):- extends(X, Z), is-a(Z, Y).
is-a(X, Y):- interface(Y), implements(X, Y).
%Facts
```

extends(mylist, arraylist).

extends(arraylist, abstractlist).

Prolog Program

A collection of facts and rules

- universal conditional statements
- ▶ Next topic

Demo time! Go to SWISH

rowable).

ble).

st, list).

serializable).

Prolog: Program Examples

is-a(X, Y):=extends(X, Y).

extends(mylist, arraylist).

COSC 221 (yong.gao@ubc.ca, UBC Okanagan) (10 / 13)

%Rules

%Facts



% A Mini Knowledgebase about Java Classes

is-a(X, Y):- extends(X, Z), is-a(Z, Y).
is-a(X, Y):- interface(Y), implements(X, Y).

```
extends(arraylist, abstractlist).
extends(error, throwable).
extends(exception, throwable).

implements(abstractlist, list).
implements(throwable, serializable).
interface(list).
```

Prolog: Program Examples - Sudoku



```
%Program Example: sudoku with CLP. It solves every
%soduko puzzle and to each puzzle, it can find all possible
%solutions. (Basic idea from the SWI-Prolog Manual)
:- use_module(library(clpfd)). %import the CLP module
board([[_,_,3,_,_,_,_].
       [....3..8.5].
       [ . .1. .2. . . ].
       [\_,\_,\_,5,\_,7,\_,\_,\_]
       [-,-,4,-,-,1,-,-]
       [_,9,_,_,_,_,],
       [5, \_, \_, \_, \_, \_, 7, 3]
       [-,-,2,-,1,-,-,-]
       [\_,\_,\_,\_,4,\_,\_,6]]).
```

Sudoku Example Cont'd



```
% the predict to solve soduko
solve:-
  board(Rows),
  length(Rows, 9), maplist(length_(9), Rows),
  append(Rows, Vs), Vs ins 1..9,
   sudoku(Rows),
   %solve the puzzle by labelling the variables
   label(Vs).
  %output the solution
  maplist(writeln, Rows).
```

References I

