UBC Department of Computer Science
Undergraduate Events
More details @ https://my.cs.ubc.ca/students/development/events

Visier Info Session

Tues., Nov 3
12 – 1:30 pm
Kaiser 2020/2030

E-Portfolio Competition Info & Training Session

Wed., Nov 4
5:45 – 7:15 pm
DMP 310

Rakuten Info Session

Thurs., Nov 5
5:30 – 6:30 pm
DMP 110

Tri-mentoring/Townhall Event

Tues., Nov 10
5:45 – 7:30 pm
X860, ICICS/CS

Tableau Open House

Thurs., Nov 12
5 – 7 pm
http://waterviewvancouver.com

Intelligent Systems (AI-2)

Computer Science cpsc422, Lecture 22

Nov, 2, 2015

Slide credit: some from Prof. Carla P. Gomes (Cornell)
some slides adapted from Stuart Russell (Berkeley), some from
Prof. Jim Martin (Univ. of Colorado)

Lecture Overview

- **Finish SAT : example**
- First Order Logics
  - Language and Semantics
  - Inference

Satisfiability problems (SAT)

Consider a CNF sentence, e.g.,

$$(\neg D \vee \neg B \vee C) \wedge (B \vee \neg A \vee \neg C) \wedge (\neg C \vee \neg B \vee E) \wedge (E \vee \neg D \vee B) \wedge (B \vee E \vee \neg C)$$

Is there an interpretation in which this sentence is true (i.e., that is a model of this sentence)?

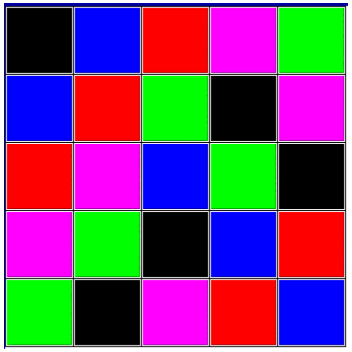
Many combinatorial problems can be reduced to checking the satisfiability of propositional sentences ……and returning a model

Encoding the Latin Square Problem in Propositional Logic

In combinatorics and in experimental design, a **Latin square** is an *n* × *n* array filled with *n* different symbols, each occurring exactly once in each row and exactly once in each column. Here is an example:

A	B	C
C	A	B
B	C	A

Here is another one:



⬛	🟦	🟥	🟪	🟩
🟦	🟥	🟩	⬛	🟪
🟥	🟪	🟦	🟩	⬛
🟪	🟩	⬛	🟦	🟥
🟩	⬛	🟪	🟥	🟦

Encoding Latin Square in Propositional Logic

Variables must be binary! (They must be propositions)

Each variables represents a color assigned to a cell.

Assume colors are encoded as integers

	1	2	3	4	...
1	⬛	🟦	🟥	🟪	🟩
2	🟦	🟥	🟩	⬛	🟪
3	🟥	🟪	🟦	🟩	⬛
4	🟪	🟩	⬛	🟦	🟥
⋮	🟩	⬛	🟪	🟥	🟦

$$x_{ijk} \in \{0,1\}$$

Assuming colors are encoded as follows

(black, 1) (red, 2) (blue, 3) (green, 4) (purple, 5)

$x_{234} = 1$

$x_{233} = 0$ True or [✓ false], ie. 0 or 1 with respect to the interpretation represented by the picture?

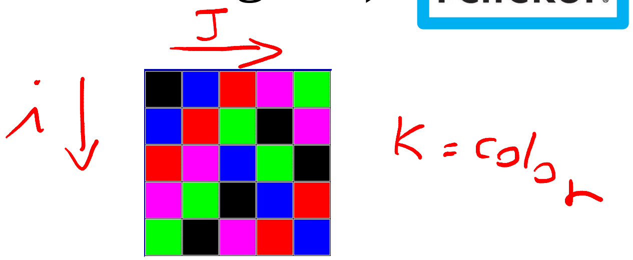
How many vars/propositions overall? n^3

Encoding Latin Square in Propositional Logic: Clauses

- Some color must be assigned to each cell (clause of length n);

$$\forall_{ij}(x_{ij1} \vee x_{ij2} \ldots x_{ijn}) \qquad \forall_{ik}(x_{i1k} \vee x_{i2k} \ldots x_{ink})$$

A. B.



⬛	🟦	🟥	🟪	🟩
🟦	🟥	🟩	⬛	🟪
🟥	🟪	🟦	🟩	⬛
🟪	🟩	⬛	🟦	🟥
🟩	⬛	🟪	🟥	🟦

- No color is repeated in the same row (sets of negative binary clauses);

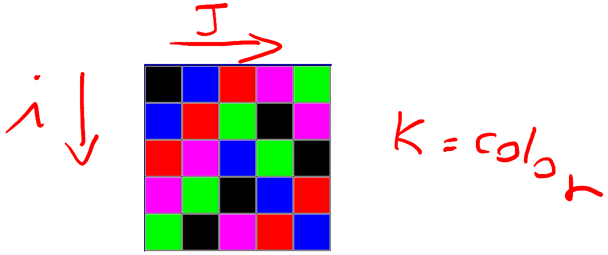
$$\forall_{ik}(\neg x_{i1k} \vee \neg x_{i2k}) \wedge (\neg x_{i1k} \vee \neg x_{i3k}) \ldots (\neg x_{i1k} \vee \neg x_{ink}) \cdots (\neg x_{ink} \vee \neg x_{i(n-1)k})$$

How many clauses?

Encoding Latin Square in Propositional Logic: Clauses

- Some color must be assigned to each cell (clause of length n);

$$\forall_{ij}(x_{ij1} \vee x_{ij2} \ldots x_{ijn})$$



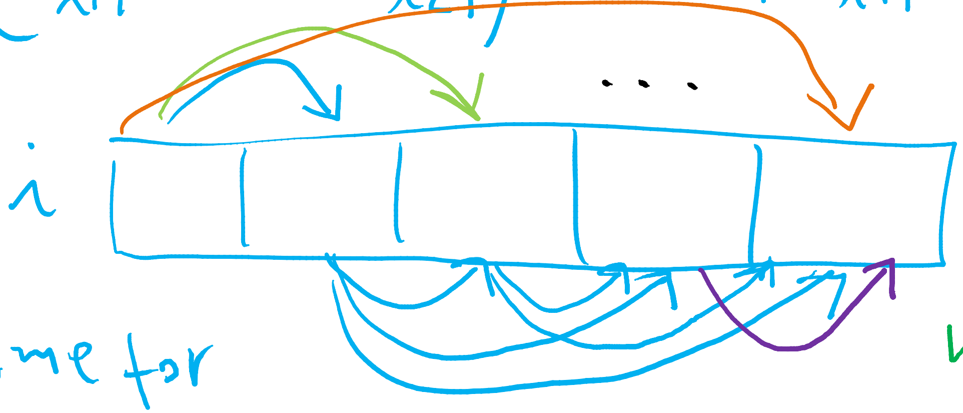
- No color is repeated in the same row (sets of negative binary clauses);

$$\forall_{ik}(\neg x_{i1k} \vee \neg x_{i2k}) \wedge (\neg x_{i1k} \vee \neg x_{i3k}) \ldots (\neg x_{i1k} \vee \neg x_{ink}) \cdots (\neg x_{i(n-1)k} \vee \neg x_{ink})$$

$n * n$

n^2

$(X_{i11} \wedge X_{i21}) \Rightarrow \neg X_{i11} \vee \neg X_{i21}$



$\frac{n * n-1}{2}$

same for

How many clauses?

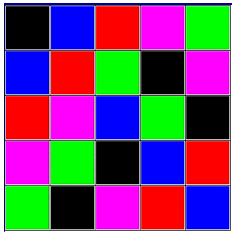
$n^2 * \frac{n * n-1}{2}$ $O(n^4)$

Encoding Latin Square Problems in Propositional Logic: FULL MODEL

n^3

Variables: x_{ijk} $cell\ i,j\ has\ color\ k;\ \ i,j,k=1,2,\ ...,n.$ $x_{ijk} \in \{0,1\}$

Each variables represents a color assigned to a cell.



Clauses: $O(n^4)$

- Some color must be assigned to each cell (clause of length n);

$$\forall_{ij}(x_{ij1} \vee x_{ij2} \ldots x_{ijn})$$

- No color is repeated in the same row (sets of negative binary clauses);

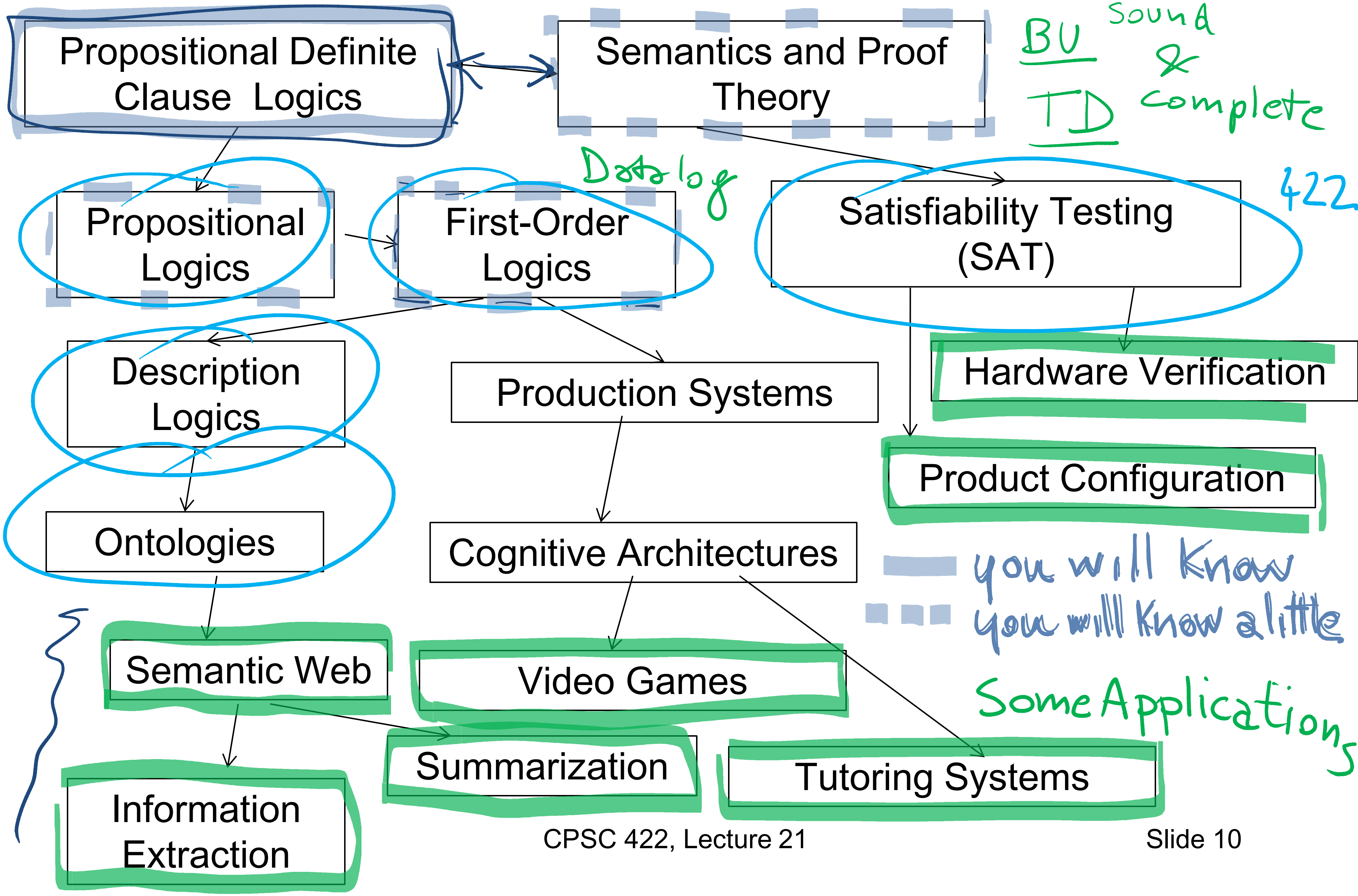
$$\forall_{ik}(\neg x_{i1k} \vee \neg x_{i2k}) \wedge (\neg x_{i1k} \vee \neg x_{i3k}) \ldots (\neg x_{i1k} \vee \neg x_{ink}) \cdots (\neg x_{i(n-1)k} \vee \neg x_{ink})$$

- No color is repeated in the same column (sets of negative binary clauses);

$$\forall_{jk}(\neg x_{1jk} \vee \neg x_{2jk}) \wedge (\neg x_{1jk} \vee \neg x_{3jk}) \ldots (\neg x_{1jk} \vee \neg x_{njk}) \cdots (\neg x_{(n-1)jk} \vee \neg x_{njk})$$

same as for rows

Logics in AI: Similar slide to the one for planning



Propositional Definite Clause Logics

Semantics and Proof Theory

BU

TD

Sound & complete

Propositional Logics

First-Order Logics

Datalog

Satisfiability Testing (SAT)

422

Description Logics

Production Systems

Hardware Verification

Product Configuration

Ontologies

Cognitive Architectures

▬ you will know

▪▪▪ you will know a little

Some Applications

Semantic Web

Video Games

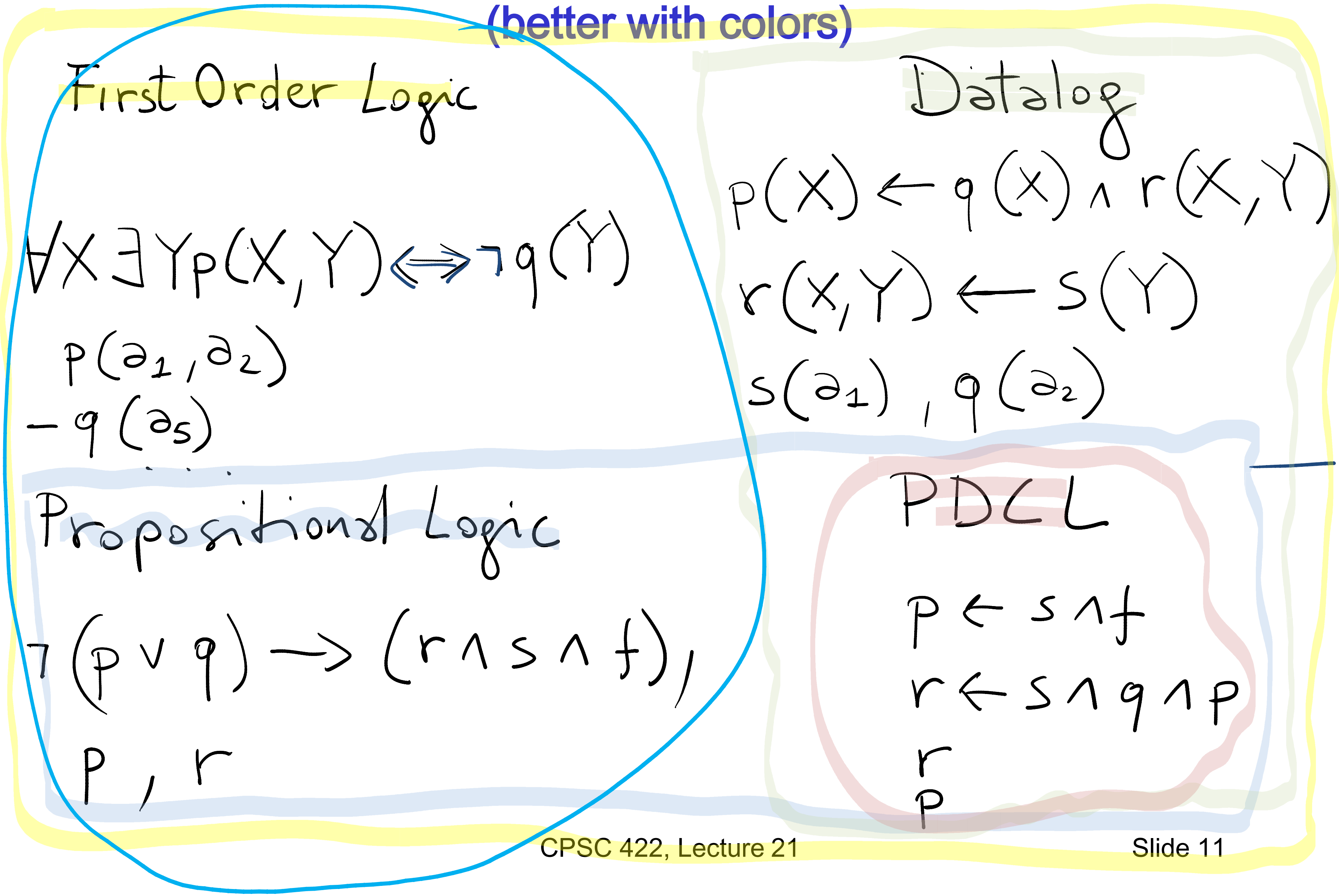
Summarization

Tutoring Systems

Information Extraction

Relationships between different Logics

(better with colors)



First Order Logic

$\forall X \exists Y p(X,Y) \Leftrightarrow \neg q(Y)$

$p(a_1,a_2)$

$-q(a_5)$

. . .

Propositional Logic

$\neg(p \vee q) \rightarrow (r \wedge s \wedge f)$,

p, r

Datalog

$p(X) \leftarrow q(X) \wedge r(X,Y)$

$r(X,Y) \leftarrow s(Y)$

$s(a_1)$, $q(a_2)$

PDCL

$p \leftarrow s \wedge f$

$r \leftarrow s \wedge q \wedge p$

r

p

Lecture Overview

- Finish SAT (example)
- **First Order Logics**
  - Language and Semantics
  - Inference

Representation and Reasoning in Complex domains (from 322)

- In complex domains expressing knowledge with **propositions** can be quite limiting

p up_s_2
q up_s_3
r ok_cb_1
s ok_cb_2
$live\_w_1$
$connected\_w_1\_w_2$

- It is often natural to consider **individuals** and their **properties**

$up(s_2)$
$up(s_3)$
$ok(cb_1)$
$ok(cb_2)$
$live(w_1)$
$connected(w_1, w_2)$

There is no notion that the system can reason about

up_s_2
up_s_3

up are about the same property

$live\_w_1$
$connected\_w_1\_w_2$

w_1 are about the same individual

(from 322) What do we gain….

By breaking propositions into relations applied to individuals?

- Express **knowledge** that **holds for set of individuals** (by introducing variables)

  $$live(W) \leftarrow connected\_to(W,W1) \land live(W1) \land wire(W) \land wire(W1).$$

- We can **ask generic queries** (i.e., containing vars)

  variabless

  $$?\ connected\_to(W, w_1)$$

“Full” First Order Logics (FOL)

LIKE DATALOG: Whereas propositional logic assumes the world contains facts, FOL (like natural language) assumes the world contains

- Objects: people, houses, numbers, colors, baseball games, wars, …
- Relations: red, round, prime, brother of, bigger than, part of, comes between, …
- Functions: father of, best friend, one more than, plus, …

FURTHERMORE WE HAVE

- More Logical Operators:….
- Equality: coreference (two terms refer to the same object)
- Quantifiers
  - ✓Statements about unknown objects
  - ✓Statements about classes of objects

Syntax of FOL

Constants	KingJohn, 2, ,...
Predicates	Brother, >,...
Functions	Sqrt, LeftLegOf,...
Variables	x, y, a, b,...
Connectives	$\neg$, $\Rightarrow$, $\wedge$, $\vee$, $\Leftrightarrow$
Equality	=
Quantifiers	$\forall$, $\exists$

Atomic sentences

Term is a *function* ($term_1$,...,$term_n$) or *constant* or *variable*

Atomic sentence is *predicate* ($term_1$,...,$term_n$) or $term_1 = term_2$

E.g.,

predicate(constant, constant)

- *Brother(KingJohn, RichardTheLionheart)*

predicate(function(function(constant)), (function(function(constant)))

- *> (Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))*

Complex sentences

Complex sentences are made from atomic sentences using connectives

$$\neg S, \quad S_1 \wedge S_2, \quad S_1 \vee S_2, \quad S_1 \Rightarrow S_2, \quad S_1 \Leftrightarrow S_2,$$

E.g.

$Sibling(KingJohn,Richard) \Rightarrow Sibling(Richard,KingJohn)$

Truth in first-order logic

Like in Prop. Logic a sentences is true with respect to an interpretation

$\neg A \wedge (B \Rightarrow C)$, 2^3

A	B	C	
T	F	F	✗
F	T	T	✓

In FOL interpretations are **much more complex** but still same idea: possible configuration of the world

symbols: { CONSTANTS → objects
Predicates → relations
Functions → functions }

2 objects △ □

2 CONSTANT SYMBOLS $\{C_1\ C_2\}$ → C_1 → △, C_2 → □

1 unary Predicate P → {△}

1 binary Predicate Q → {{△, △}}

IS ∀x P(x) TRUE?

A. yes
B. no

iclicker.

Truth in first-order logic

Like in Prop. Logic a sentences is true with respect to an interpretation

$\neg A \wedge B \Rightarrow C,$ 2^3

A	B	C	
T	F	F	✗
T	T	T	✓

In FOL interpretations are much more complex but still same idea: possible configuration of the world

symbols: { CONSTANTS → objects; Predicates → relations; Functions → functions }

2 objects △ □

2 CONSTANT SYMBOLS $\{c_1\ c_2\}$ → $c_1 \rightarrow$ △ $c_2 \rightarrow$ □

1 unary Predicate P → {△}

1 binary Predicate Q → {{△, △}}

IS $\forall x\ P(x)$ TRUE? NO

but if P → {△, □} Yes!

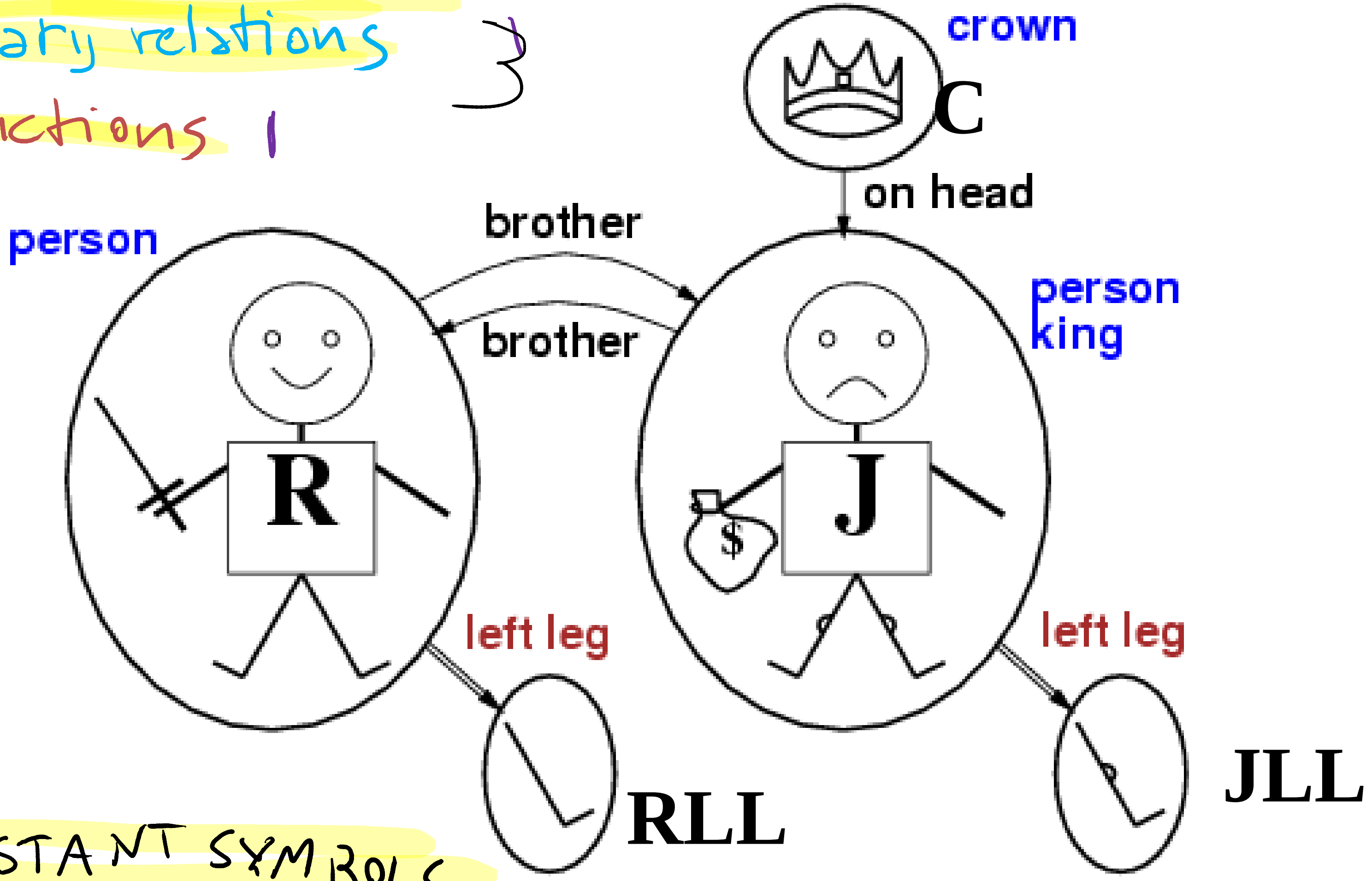
Interpretations for FOL: Example

5 objects

binary relations 2

unary relations 3

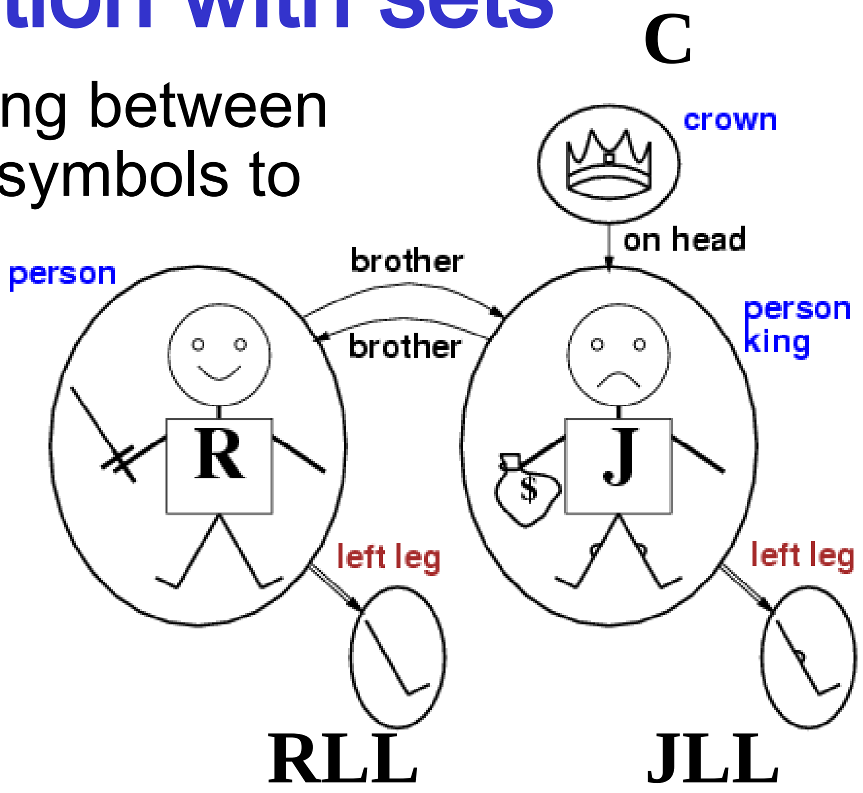
functions 1



CONSTANT SYMBOLS 5

Same interpretation with sets

Since we have a one to one mapping between symbols and object we can use symbols to refer to objects



- {R, J, RLL, JLL, C}

Property Predicates

- Person = {R, J}
- Crown = {C}
- King = {J}

Relational Predicates

- Brother = { <R,J>, <J,R>}
- OnHead = {<C,J>}

Functions

- LeftLeg = {<R, RLL>, <J, JLL>}

How many Interpretations with….

assuming unique names

5 Objects and 5 symbols
- {R, J, RLL, JLL, C}

$5!$

J R C RLL JLL

3 Property Predicates (Unary Relations)
- Person
- Crown
- King

R J RLL JLL C

0/1 0/1 0/1 0/1 0/1

2^5

A. 2^5 B. 2^{25} C. 25^2

2 Relational Predicates
- Brother
- OnHead

25 possibilities; each one can be 0/1 so 2^{25}

1 Function
- LeftLeg

5^5

TOTAL $5! * (2^5)^3 * (2^{25})^2 * 5^5$

To summarize: Truth in first-order logic

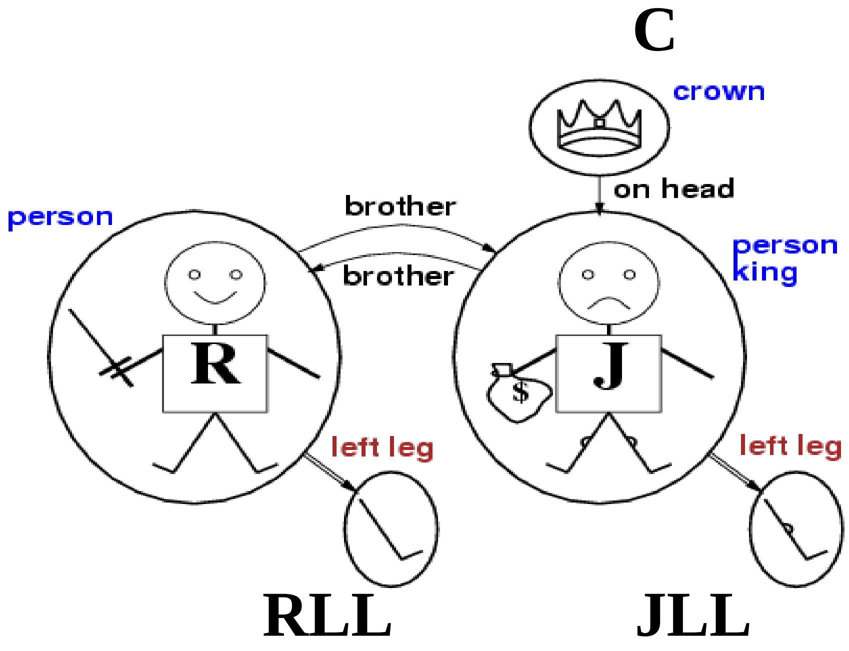
- Sentences are true with respect to an **interpretation**
- World contains objects (**domain elements**)
- Interpretation specifies referents for

constant symbols	→	**objects**
predicate symbols	→	**relations**
function symbols	→	**functional relations**

- An atomic sentence ***predicate(term$_1$,...,term$_n$)*** is true

  iff the **objects** referred to by ***term$_1$,...,term$_n$***

  are in the **relation** referred to by ***predicate***



Quantifiers

Allows us to express

- **Properties of collections of objects** instead of enumerating objects by name
- **Properties of an unspecified object**

Universal: “for all” $\forall$

Existential: “there exists” $\exists$

Universal quantification

$\forall$<*variables*> <*sentence*>

Everyone at UBC is smart:

$\forall x\ At(x, UBC) \Rightarrow Smart(x)$

$\forall x$ *P* is true in an interpretation ***I*** iff *P* is true with *x* being each possible object in ***I***

Equivalent to the conjunction of instantiations of *P*

$At(KingJohn, UBC) \Rightarrow Smart(KingJohn)$

$\wedge At(Richard, UBC) \Rightarrow Smart(Richard)$

$\wedge At(Ralphie, UBC) \Rightarrow Smart(Ralphie)$

$\wedge \ldots$

Existential quantification

$\exists$<*variables*> <*sentence*>

Someone at UBC is smart:

$\exists x$ At(x, UBC) $\wedge$ Smart(x)

$\exists x\ P$ is true in an interpretation ***I*** iff P is true with x being some possible object in ***I***

Equivalent to the disjunction of instantiations of P

At(KingJohn, UBC) $\wedge$ Smart(KingJohn)

$\vee$ At(Richard, UBC) $\wedge$ Smart(Richard)

$\vee$ At(Ralphie, UBC) $\wedge$ Smart(Ralphie)

$\vee$ ...

Properties of quantifiers

$\exists x\ \forall y$ is not the same as $\forall y\ \exists x$

$\exists x\ \forall y\ Loves(x,y)$

- "There is a person who loves everyone in the world"

$\forall y\ \exists x\ Loves(x,y)$

- "Everyone in the world is loved by at least one person"

Quantifier duality: each can be expressed using the other

$\forall x\ Likes(x,IceCream) \qquad \neg\exists x\ \neg Likes(x,IceCream)$

$\exists x\ Likes(x,Broccoli) \qquad \neg\forall x\ \neg Likes(x,Broccoli)$

Lecture Overview

- Finish SAT (example)
- **First Order Logics**
  - Language and Semantics
  - **Inference**

FOL: Inference

Resolution Procedure can be generalized to FOL

- **Every formula** can be rewritten in **logically equivalent CNF**
  - Additional rewriting rules for quantifiers
- **Similar Resolution step,** but variables need to be unified (like in DATALOG)

$In(x,y) \lor \neg Charged(x)$ $\theta = \{z/x, v/y\}$

$\neg In(z,v) \lor Connected(z)$

$\boxed{\neg Charged(x) \lor Connected(x)}$

NLP Practical Goal for FOL: the ultimate Web question-answering system?

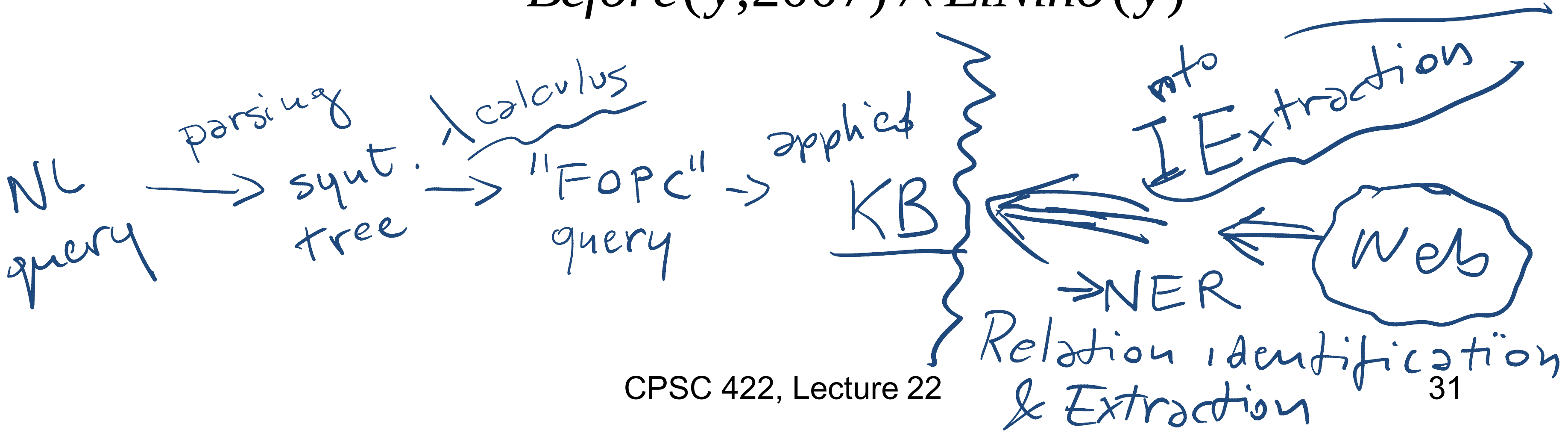
Map NL queries into FOPC so that answers can be effectively computed

What African countries are not on the Mediterranean Sea?

$$\exists c\ Country(c) \wedge \neg Borders(c, Med.Sea) \wedge In(c, Africa)$$

- *Was 2007 the first El Nino year after 2001?*

$$ElNino(2007) \wedge \neg \exists y\ Year(y) \wedge After(y,2001) \wedge Before(y,2007) \wedge ElNino(y)$$



Learning Goals for today's class

You can:

- Explain differences between Proposition Logic and First Order Logic
- Compute number of interpretations for FOL
- Explain the meaning of quantifiers
- Describe application of FOL to NLP: Web question answering

Next class Wed

- Ontologies (e.g., Wordnet, Probase), Description Logics…

Assignment-3 out today

Categories & Events

- **Categories:**
  - *VegetarianRestaurant (Joe's) - relation vs. object*
  - *MostPopular(Joe's, VegetarianRestaurant)*
  - *ISA (Joe's, VegetarianRestaurant)*
  - *AKO (VegetarianRestaurant, Restaurant)*

Reification

Events: can be described in NL with different numbers of arguments…

- *I ate*
- *I ate a turkey sandwich*
- *I ate a turkey sandwich at my desk*
- *I ate at my desk*
- *I ate lunch*
- *I ate a turkey sandwich for lunch*
- *I ate a turkey sandwich for lunch at my desk*

eating (all possible arguments)

Reification Again

and an apple

“I ate a turkey sandwich ~~for lunch~~”

∃ *w: Isa(w,Eating)* ∧ *Eater(w,Speaker)* ∧
Eaten(w,TurkeySandwich) ∧ *~~MealEaten(w,Lunch~~)* ∧
Eaten(w, apple)

Reification Advantage:

- No need to specify fixed number of arguments to represent a given sentence in NL

MUC-4 Example

On October 30, 1989, one civilian was killed in a reported FMLN attack in El Salvador.

INCIDENT: DATE	30 OCT 89
INCIDENT: LOCATION	EL SALVADOR
INCIDENT: TYPE	ATTACK
INCIDENT: STAGE OF EXECUTION	ACCOMPLISHED
INCIDENT: INSTRUMENT ID	
INCIDENT: INSTRUMENT TYPE	
PERP: INCIDENT CATEGORY	TERRORIST ACT
PERP: INDIVIDUAL ID	"TERRORIST"
PERP: ORGANIZATION ID	"THE FMLN"
PERP: ORG. CONFIDENCE	REPORTED: "THE FMLN"
PHYS TGT: ID	
PHYS TGT: TYPE	
PHYS TGT: NUMBER	
PHYS TGT: FOREIGN NATION	
PHYS TGT: EFFECT OF INCIDENT	
PHYS TGT: TOTAL NUMBER	
HUM TGT: NAME	
HUM TGT: DESCRIPTION	"1 CIVILIAN"
HUM TGT: TYPE	CIVILIAN: "1 CIVILIAN"
HUM TGT: NUMBER	1: "1 CIVILIAN"
HUM TGT: FOREIGN NATION	
HUM TGT: EFFECT OF INCIDENT	DEATH: "1 CIVILIAN"
HUM TGT: TOTAL NUMBER	

Representing Time

Events are associated with points or intervals in time.

We can impose an ordering on distinct events using the notion of *precedes*.

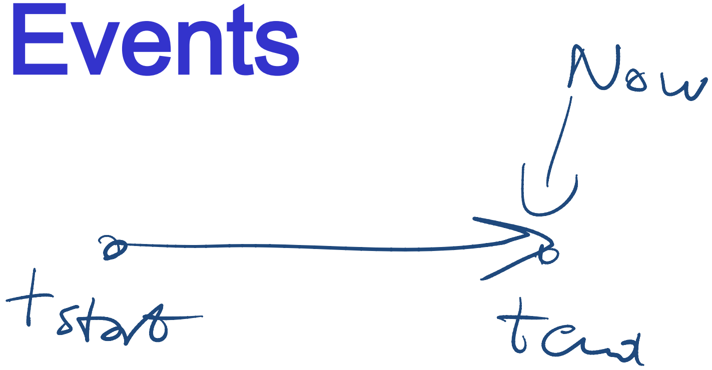
- **Temporal logic notation:**
  $(\exists w,x,t)\ \underline{Arrive(w,x,t)}$ no reification
- **Constraints on variable *t***

  I arrived in New York

  $(\exists\ t)\ Arrive(I,NewYork,t) \wedge precedes(t,Now)$

Interval Events

Need t_{start} and t_{end}



"She was driving to New York until now"

$$\exists\, t_{start}, t_{end}, e, i, x \quad \land\, female(x)$$

$$ISA(e,Drive) \land Driver(e, \overset{x}{\boxed{She}}) \land$$

$$Dest(e, NewYork) \land IntervalOf(e,i) \land$$

$$Endpoint(i, t_{end}) \land Startpoint(i, t_{start}) \land$$

$$Precedes(t_{start}, Now) \land$$

$$Equals(t_{end}, Now)$$

Review --- Syntactic Ambiguity

- FOPC provides many ways to represent the same thing.
- E.g., “Ball-5 is red.”
  - HasColor(Ball-5, Red)
    - Ball-5 and Red are objects related by HasColor.
  - Red(Ball-5)
    - Red is a unary predicate applied to the Ball-5 object.
  - HasProperty(Ball-5, Color, Red)
    - Ball-5, Color, and Red are objects related by HasProperty.
  - ColorOf(Ball-5) = Red
    - Ball-5 and Red are objects, and ColorOf() is a function.
  - HasColor(Ball-5(), Red())
    - Ball-5() and Red() are functions of zero arguments that both return an object, which objects are related by HasColor.
  - …

- This can GREATLY confuse a pattern-matching reasoner.
  - Especially if multiple people collaborate to build the KB, and they all have different representational conventions.

Review --- Syntactic Ambiguity --- Partial Solution

- FOL can be TOO expressive, can offer TOO MANY choices

- Likely confusion, especially for **teams** of Knowledge Engineers

- Different team members can make different representation choices
  - E.g., represent “Ball43 is Red.” as:
    - a predicate (= verb)? E.g., “Red(Ball43)” ?
    - an object (= noun)? E.g., “Red = Color(Ball43))” ?
    - a property (= adjective)? E.g., “HasProperty(Ball43, Red)” ?

- PARTIAL SOLUTION:
  - An upon-agreed **ontology** that settles these questions
  - Ontology = what exists in the world & how it is represented
  - The Knowledge Engineering teams agrees upon an ontology BEFORE they begin encoding knowledge

Relation Between Tenses and Time

Relation between simple verb tenses and points in time is not straightforward

Present tense used like future:

- *We fly from Baltimore to Boston at 10*

Complex tenses:

- *Flight 1902 arrived late*
- *Flight 1902 had arrived late*

Representing them in the same way seems wrong….

Reference Point

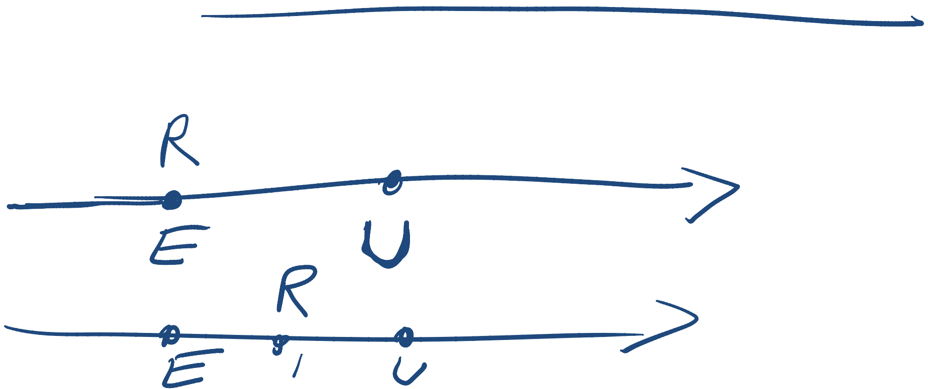
Reichenbach (1947) introduced notion of Reference point (R), separated out from Utterance time (U) and Event time (E)

Example:

R E

- *When Mary's flight departed, I ate lunch*
- *When Mary's flight departed, I had eaten lunch*

Departure event specifies reference point.



Today Feb 7

Semantics / Meaning /Meaning Representations

Linguistically relevant Concepts in FOPC / FOL

Semantic Analysis

Limited expressiveness of propositional logic

KB contains "physics" sentences for every single square

For every time *t* and every location [*x,y*],

$L_{x,y} \wedge FacingRight^t \wedge Forward^t \Rightarrow L_{x+1,y}$

Rapid proliferation of clauses.

First order logic is designed to deal with this through the introduction of variables.

Summary

Logical agents apply inference to a knowledge base to derive new information and make decisions

Basic concepts of logic:

- syntax: formal structure of sentences
- semantics: truth of sentences wrt models
- entailment: necessary truth of one sentence given another
- inference: deriving sentences from other sentences
- soundness: derivations produce only entailed sentences
- completeness: derivations can produce all entailed sentences

Resolution is complete for propositional logic

Forward, backward chaining are linear-time, complete for Horn clauses

Propositional logic lacks expressive power

Resolution Algorithm

$KB \models \alpha$ *equivalent to*

$KB \land \neg\alpha$ *unsatisfiable*

- The resolution algorithm tries to prove:
- $KB \land \neg\alpha$ is converted in CNF
- Resolution is applied to each pair of clauses with complementary literals $\neg q \lor r \lor s \quad p \lor q \rightarrow r \lor s \lor p$
- Resulting clauses are added to the set (if not already there)

- Process continues until one of two things can happen:

1. Two clauses resolve in the empty clause. i.e. query is entailed

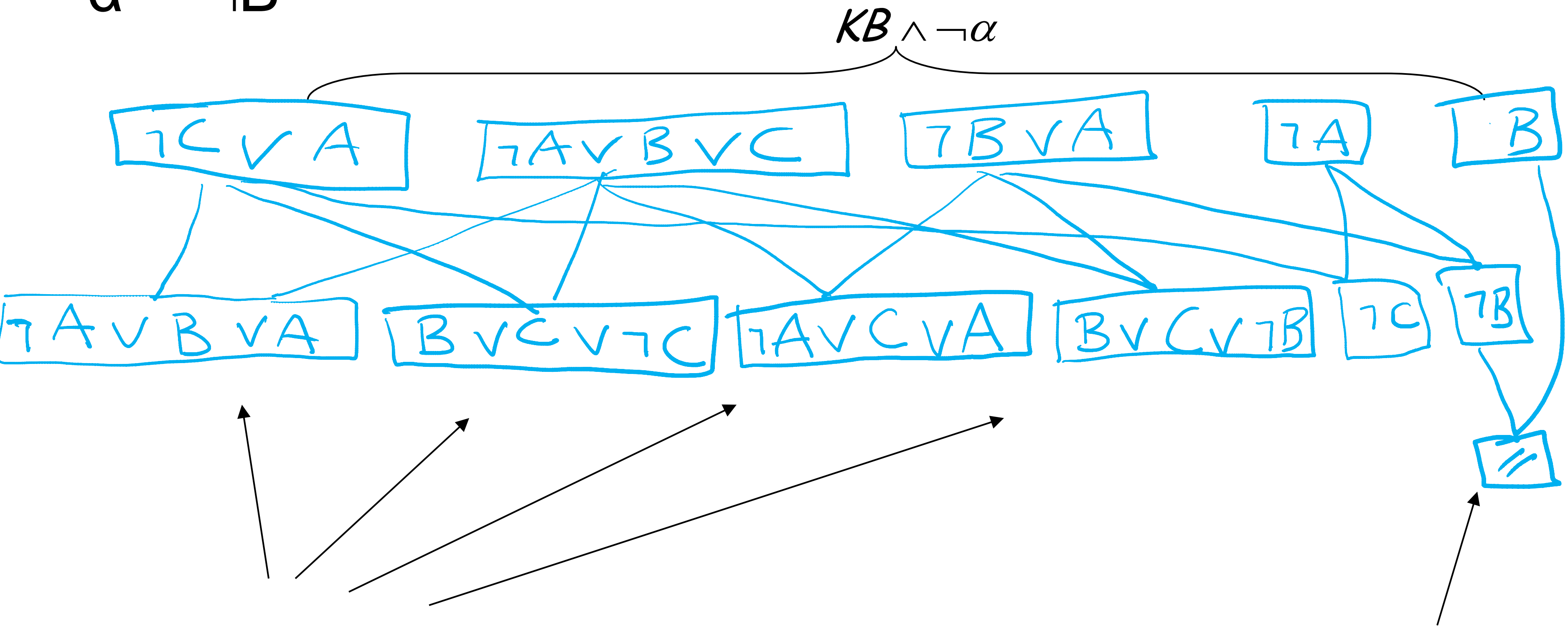
$p \quad \neg p \rightarrow \emptyset \qquad > KB \vdash_R \alpha \Rightarrow KB \models \alpha$

2. No new clauses can be added: We find no contradiction, there is a model that satisfies the sentence and hence we cannot entail the query. $KB \nvdash_R \alpha \Rightarrow KB \not\models \alpha$

Resolution example

$KB = (A \Leftrightarrow (B \vee C)) \wedge \neg A$

$\alpha = \neg B$



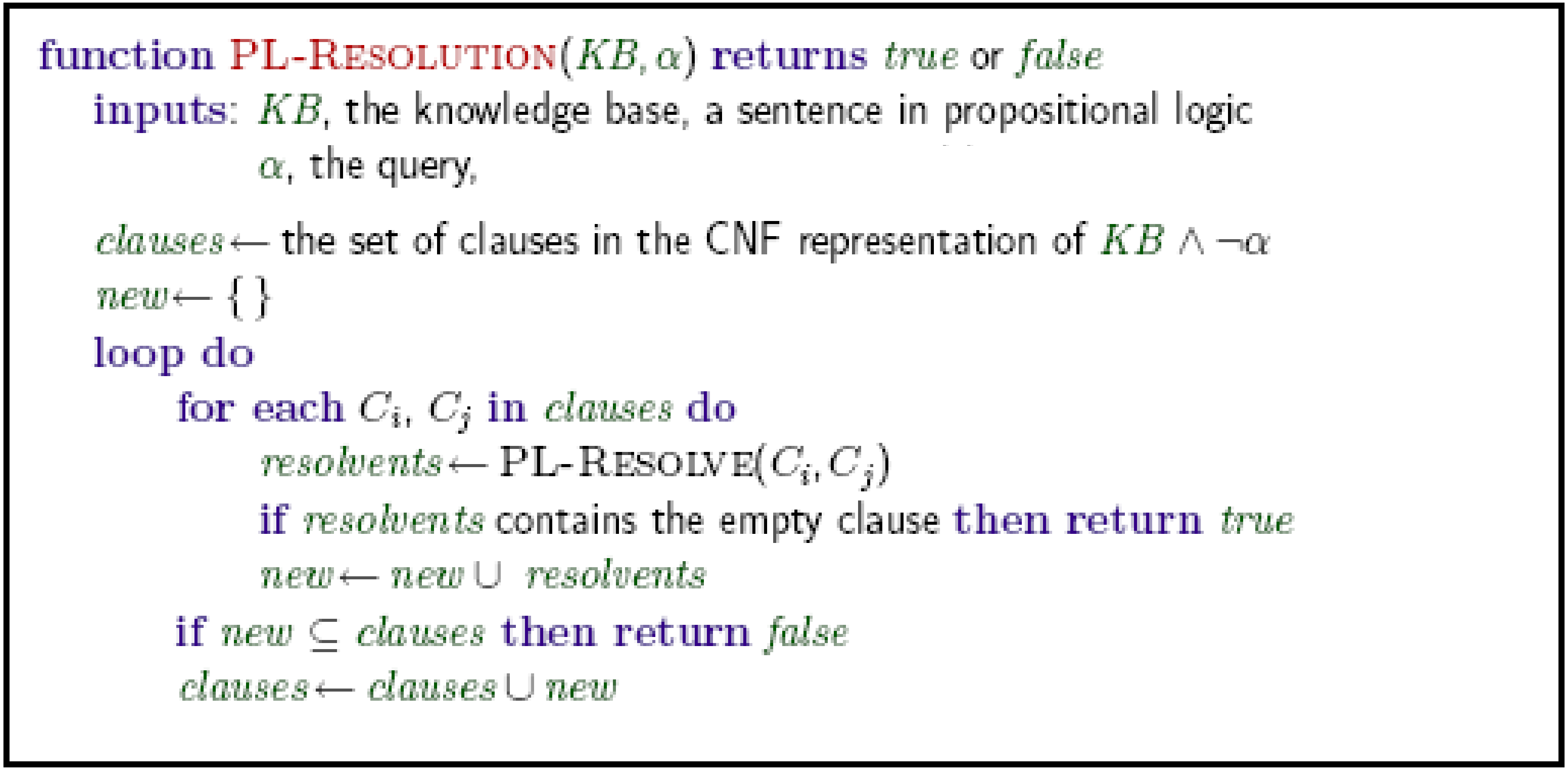
$KB \wedge \neg\alpha$

True!

False in all worlds

Resolution algorithm

Proof by contradiction, i.e., show $KB \wedge \neg\alpha$ unsatisfiable



```
function PL-RESOLUTION($KB, \alpha$) returns true or false
    inputs: $KB$, the knowledge base, a sentence in propositional logic
            $\alpha$, the query,

    clauses ← the set of clauses in the CNF representation of $KB \wedge \neg\alpha$
    new ← { }
    loop do
        for each $C_i$, $C_j$ in clauses do
            resolvents ← PL-RESOLVE($C_i, C_j$)
            if resolvents contains the empty clause then return true
            new ← new ∪ resolvents
        if new ⊆ clauses then return false
        clauses ← clauses ∪ new
```

Requirements for Meaning Representations

Answer questions Y/N

Inference

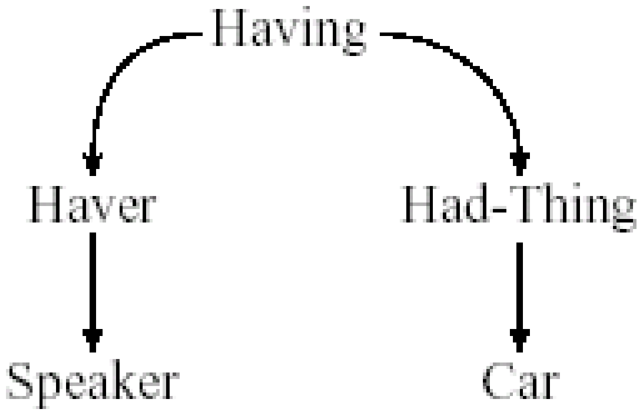
Pred Arguments

Common Meaning Representations

I have a car

$$\exists x,y\, Having(x) \wedge Haver(Speaker,x) \wedge HadThing(y,x) \wedge Car(y)$$

FOPC



Semantic Nets

Frames

```
Having
 Haver:   Speaker
 HadThing:   Car
```

Common foundation: structures composed of symbols that correspond to **objects** and **relationships**

Today Feb 7

Semantics / Meaning /Meaning Representations

Linguistically relevant Concepts in FOPC/FOL

Semantic Analysis

Linguistically Relevant Concepts in FOPC

Categories & Events (Reification)

Representing Time

Beliefs *(optional, read if relevant to your project)*

Aspects *(optional, read if relevant to your project)*

extend FOL

Description Logics *(optional, read if relevant to your project)*

restrict FOL

Semantic Web

Variables

A big part of using FOL involves keeping track of all the variables while reasoning.

Substitution lists are the means used to track the value, or binding, of variables as processing proceeds.

$$\{\mathrm{var}/\,\mathit{term}, \mathrm{var}/\,\mathit{term}, \mathrm{var}/\,\mathit{term}\ldots\}$$

Examples

$$Cat(Felix)$$

$$\forall x Cat(x) \rightarrow Annoying(x)$$

$$\{x / Felix\}$$

$$Cat(Felix) \rightarrow Annoying(Felix)$$

Examples

$$\forall x, y Near(x, y) \rightarrow Near(y, x)$$

$$\{x / McCoy, y / ChemE\}$$

$$Near(McCoy, ChemE) \rightarrow Near(ChemE, McCoy)$$

Semantics: Worlds

The world **consists of** objects **that have** properties**.**

- **There are** relations **and** functions **between these objects**
- **Objects in the world, individuals:** people, houses, numbers, colors, baseball games, wars, centuries
  - ✓ Clock A, John, 7, the-house in the corner, Tel-Aviv, Ball43
- **Functions** on individuals:
  - ✓ father-of, best friend, third inning of, one more than
- **Relations**:
  - ✓ brother-of, bigger than, inside, part-of, has color, occurred after
- **Properties (a relation of arity 1)**

Semantics: Interpretation

An **interpretation** of a sentence (wff) is an assignment that maps

- Object constant symbols to objects in the world,
- n-ary function symbols to n-ary functions in the world,
- n-ary relation symbols to n-ary relations in the world

Given an interpretation, an atomic sentence has the value “true” if it denotes a relation that holds for those individuals denoted in the terms. Otherwise it has the value “false.”

Semantics: Models

An interpretation satisfies a wff (sentence) if the wff has the value “true” under the interpretation.

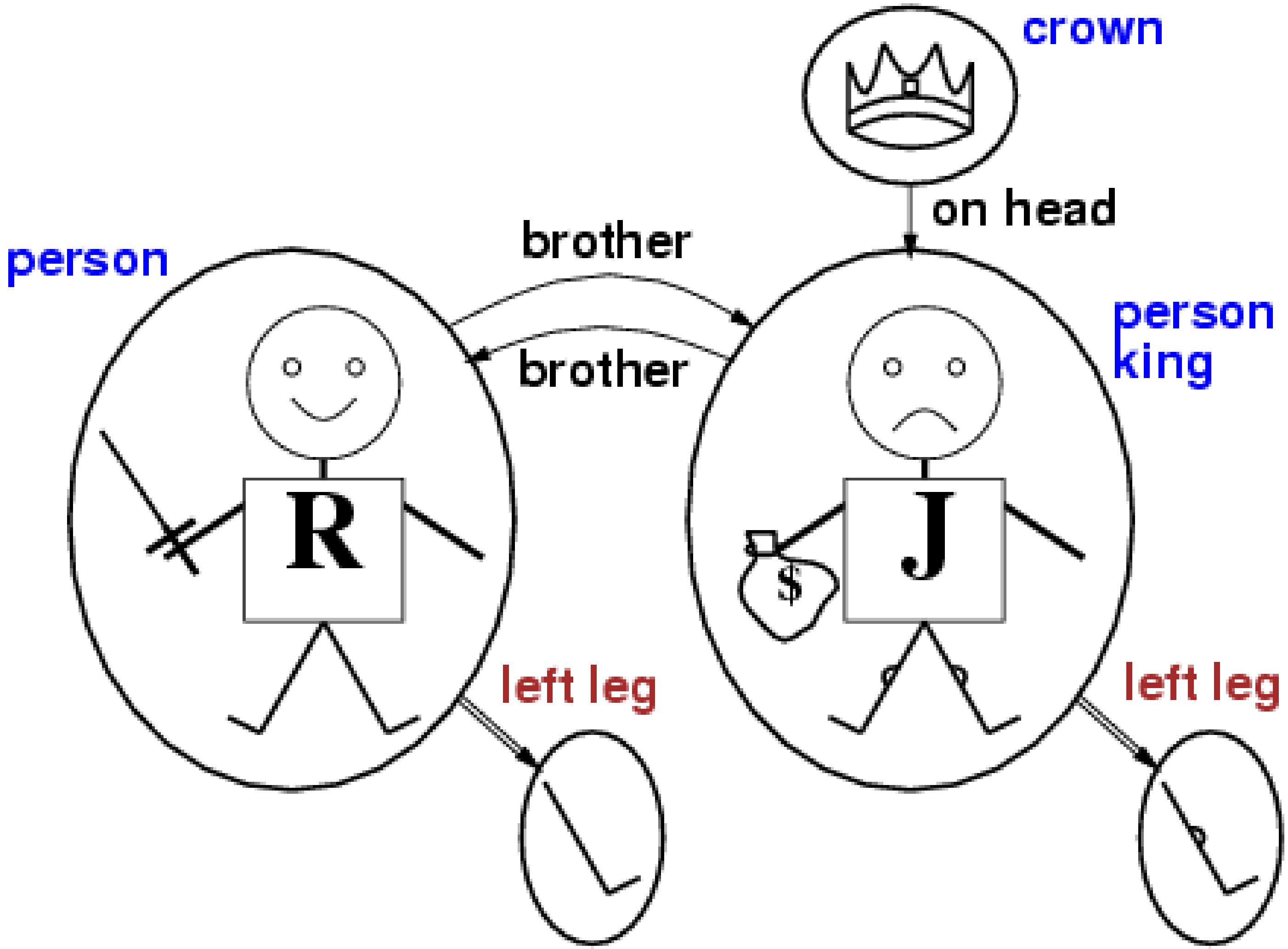
Model: A domain and an interpretation that satisfies a wff is a model of that wff

Validity: Any wff that has the value “true” under all interpretations is valid

Any wff that does not have a model is inconsistent or unsatisfiable

If a wff w has a value true under all the models of a set of sentences KB then KB logically entails w

Interpretations for FOL: Example



Pros and cons of propositional logic

☺ Propositional logic is declarative

☺ Propositional logic allows partial/disjunctive/negated information

- (unlike most data structures and databases)

☺ Propositional logic is compositional:

- meaning of $B_{1,1} \wedge P_{1,2}$ is derived from meaning of $B_{1,1}$ and of $P_{1,2}$
-

☺ Meaning in propositional logic is context-independent

- (unlike natural language, where meaning depends on context)
-

☹ Propositional logic has very limited expressive power

- (unlike natural language)
- E.g., cannot say "pits cause breezes in adjacent squares“
  - ✓except by writing one sentence for each square