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

Salesforce Info Session

Mon., Oct 26
6 – 7 pm
DMP 310

Dynastream Info Session

Thurs., Oct 29
5:30 – 6:30 pm
DMP 110

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

Intelligent Systems (AI-2)

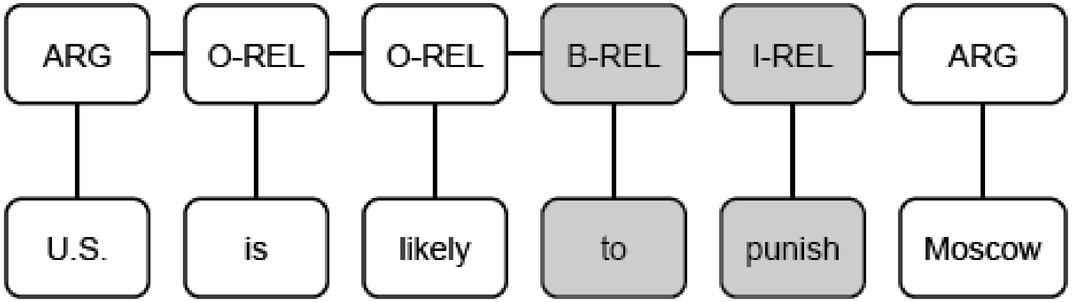
Computer Science cpsc422, Lecture 20

Oct, 28, 2015

Slide credit: some slides adapted from Stuart Russell (Berkeley), some from Padhraic Smyth (UCIrvine)

PhD thesis I was reviewing some months ago…
University of Alberta
EXTRACTING INFORMATION NETWORKS FROM TEXT

We model *predicate detection* as a sequence labeling problem — …. We adopt the BIO encoding, a widely-used technique in NLP.
Our method, called Meta-CRF, is based on Conditional Random Fields (CRF) .
CRF is a graphical model that estimates a conditional probability distribution, denoted p(yjx), over label sequence y given the token sequence x.



422 big picture: Where are we?

Hybrid: Det +Sto

Prob CFG
Prob Relational Models
Markov Logics

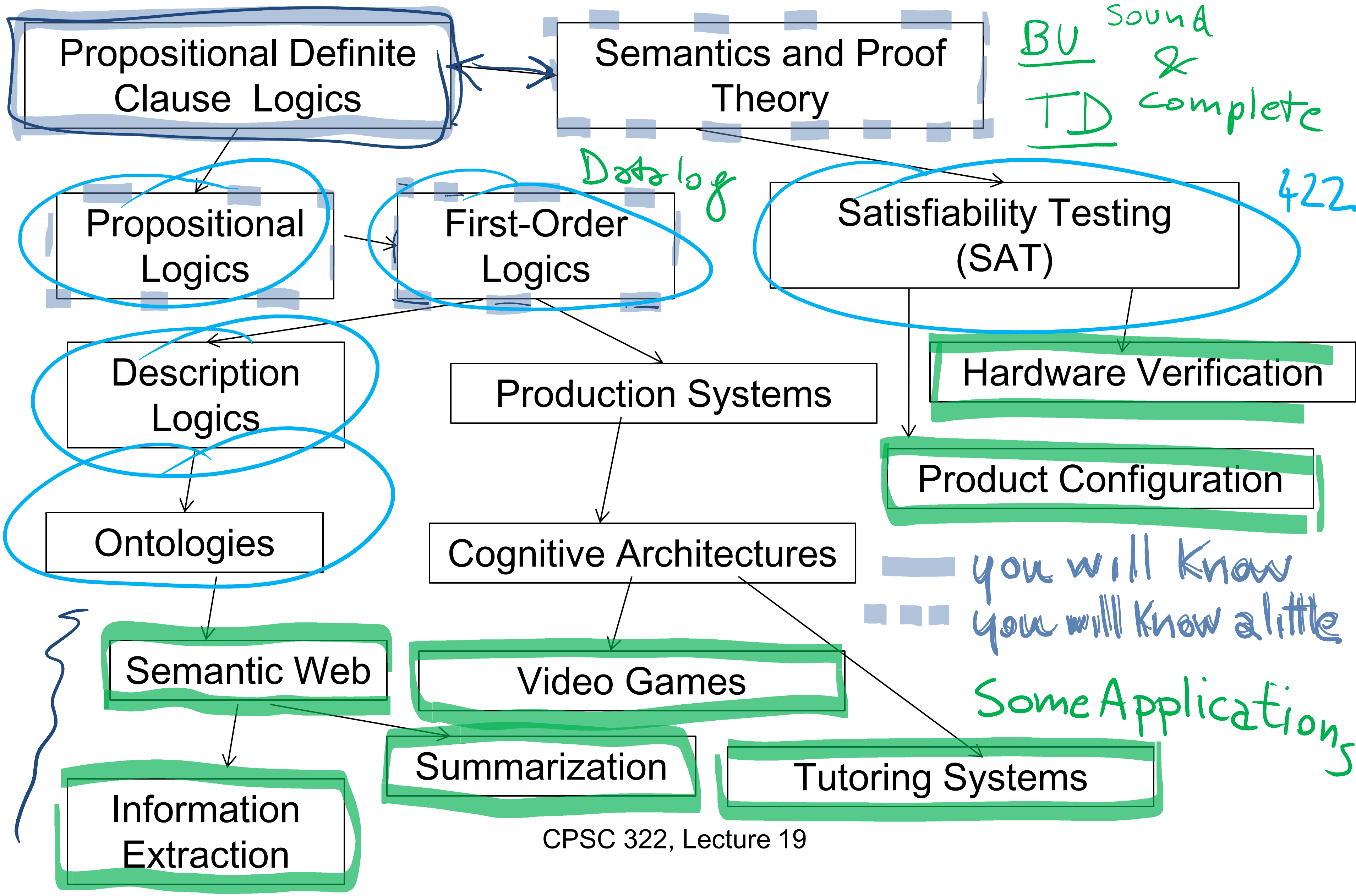
	Deterministic	Stochastic
Query	*Logics* *First Order Logics* *Ontologies* *Temporal rep.* • Full Resolution • SAT	*Belief Nets* Approx. : Gibbs *Markov Chains and HMMs* Forward, Viterbi…. Approx. : Particle Filtering *Undirected Graphical Models* *Markov Networks* *Conditional Random Fields*
Planning		*Markov Decision Processes and Partially Observable MDP* • Value Iteration • Approx. Inference *Reinforcement Learning*

Applications of AI

Representation

Reasoning Technique

Logics in AI: Similar slide to the one for planning



Relationships between different Logics

(better with colors)

First Order Logic

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

$p(a_1, a_2)$

$\neg q(a_5)$

...

Datalog

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

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

$s(a_1), q(a_2)$

Propositional Logic

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

p, r

PDCL

$p \leftarrow s \wedge f$

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

r

p

Lecture Overview

- **Basics Recap: Interpretation / Model /..**
- Propositional Logics
- Satisfiability, Validity
- Resolution in Propositional logics

Basic definitions from 322 (Semantics)

Definition (interpretation)
An interpretation I assigns a truth value to each atom.

Definition (truth values of statements cont'**):** A knowledge base *KB* is true in I if and only if every clause in *KB* is true in I.

PDC Semantics: Knowledge Base (KB)

- A **knowledge base KB** is true in I if and only if every clause in KB is true in I.

	p	q	r	s
I_1	*true*	*true*	*false*	*false*

i>clicker®

Which of the three KB below is True in I_1 ?

A

p
r
$s \leftarrow q \wedge p$

B

p
q
$s \leftarrow q$

C

p
$q \leftarrow r \wedge s$

PDC Semantics: Knowledge Base (KB)

- A **knowledge base KB** is true in I if and only if every clause in KB is true in I.

	p	q	r	s
I_1	*true*	*true*	*false*	*false*

KB_1

p
r
$s \leftarrow q \wedge p$

KB_2

p
q
$s \leftarrow q$

KB_3

p
$q \leftarrow r \wedge s$

Which of the three KB above is True in I_1 ? ***KB_3***

Basic definitions from 322 (Semantics)

Definition (interpretation)
An interpretation I assigns a truth value to each atom.

Definition (truth values of statements cont'**):** A knowledge base *KB* is true in I if and only if every clause in *KB* is true in I.

Definition (model)
A model of a set of clauses (a KB) is an interpretation in which all the clauses are *true*.

Example: Models

$$KB = \begin{cases} p \leftarrow q. \\ q. \\ r \leftarrow s. \end{cases}$$

	p	q	r	s
I_1	*true*	*true*	*true*	*true*
I_2	*false*	*false*	*false*	*false*
I_3	*true*	*true*	*false*	*false*
I_4	*true*	*true*	*true*	*false*
I_5	*true*	*true*	*false*	*true*

Which interpretations are models?

Basic definitions from 322 (Semantics)

Definition (interpretation)
An interpretation I assigns a truth value to each atom.

Definition (truth values of statements cont'**):** A knowledge base *KB* is true in I if and only if every clause in *KB* is true in I.

Definition (model)
A model of a set of clauses (a KB) is an interpretation in which all the clauses are *true*.

Definition (logical consequence)
If *KB* is a set of clauses and *G* is a conjunction of atoms, *G* is a logical consequence of *KB*, written $KB \models G$, if *G* is *true* in every model of *KB*.

iclicker.

Is it true that if

$M(KB)$ is the set of all models of KB

$M(\alpha)$ is the set of all models of α

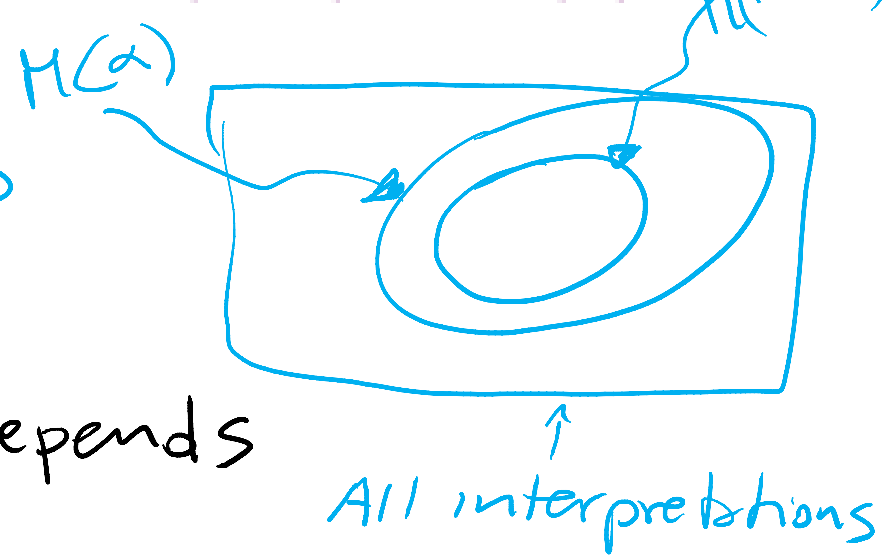
Then $KB \models \alpha$ if and only if $M(KB) \subseteq M(\alpha)$

α true in all the models of KB

A. yes

B. no

C. It depends



Basic definitions from 322 (Proof Theory)

Definition (soundness)

A proof procedure is sound if $KB \vdash G$ implies $KB \models G$.

Definition (completeness)

A proof procedure is complete if $KB \models G$ implies $KB \vdash G$.

Lecture Overview

- Basics Recap: Interpretation / Model /
- **Propositional Logics**
- Satisfiability, Validity
- Resolution in Propositional logics

Relationships between different Logics

(better with colors)

First Order Logic

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

$p(a_1, a_2)$

$-q(a_5)$

...

Datalog

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

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

$s(a_1), q(a_2)$

Propositional Logic

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

p, r

PDCL

$p \leftarrow s \wedge f$

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

r

p

Propositional logic: Syntax

Atomic sentences = single proposition symbols

- E.g., P, Q, R
- Special cases: True = always true, False = always false

Complex sentences:

- If S is a sentence, $\neg$S is a sentence (negation)
- If S_1 and S_2 are sentences, $S_1 \wedge S_2$ is a sentence (conjunction)
- If S_1 and S_2 are sentences, $S_1 \vee S_2$ is a sentence (disjunction)
- If S_1 and S_2 are sentences, $S_1 \Rightarrow S_2$ is a sentence (implication)
- If S_1 and S_2 are sentences, $S_1 \Leftrightarrow S_2$ is a sentence (biconditional)

Propositional logic: Semantics

Each interpretation specifies true or false for each proposition symbol

E.g.	p	q	r
	false	true	false

Rules for evaluating truth with respect to an interpretation I :

$\neg S$	is true iff	S is false	
$S_1 \wedge S_2$	is true iff	S_1 is true and	S_2 is true
$S_1 \vee S_2$	is true iff	S_1 is true or	S_2 is true
$S_1 \Rightarrow S_2$	is true iff	S_1 is false or	S_2 is true
i.e.,	is false iff	S_1 is true and	S_2 is false
$S_1 \Leftrightarrow S_2$	is true iff	$S_1 \Rightarrow S_2$ is true and $S_2 \Rightarrow S_1$ is true	

Simple recursive process evaluates an arbitrary sentence, e.g.,

$(\neg p \wedge (q \vee r)) \Leftrightarrow \neg p$ = $(\neg F \wedge (T \vee F)) \Leftrightarrow \neg F$

$(T \wedge T) \Leftrightarrow T$

$T \Leftrightarrow T$

Logical equivalence

Two sentences are logically equivalent iff true in same interpretations

$\alpha \equiv \beta$ if and only if $\alpha \models \beta$ and $\beta \models \alpha$

they have the same models

$(\alpha \wedge \beta)$	$\equiv$	$(\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta)$	$\equiv$	$(\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma)$	$\equiv$	$(\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma)$	$\equiv$	$(\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha)$	$\equiv$	α	double-negation elimination
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta)$	$\equiv$	$((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta)$	$\equiv$	$(\neg\alpha \vee \neg\beta)$	De Morgan
$\neg(\alpha \vee \beta)$	$\equiv$	$(\neg\alpha \wedge \neg\beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma))$	$\equiv$	$((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma))$	$\equiv$	$((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha) \equiv \alpha$	double-negation elimination
$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$	De Morgan
$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

Can be used to rewrite formulas….

$(p \Rightarrow \neg(q \wedge r))$

$\neg p \vee \neg(q \wedge r)$

$\neg p \vee \neg q \vee \neg r$

$(\alpha \wedge \beta)$	$\equiv$	$(\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta)$	$\equiv$	$(\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma)$	$\equiv$	$(\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma)$	$\equiv$	$(\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha)$	$\equiv$	α	double-negation elimination
* $(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta)$	$\equiv$	$((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta)$	$\equiv$	$(\neg\alpha \vee \neg\beta)$	De Morgan
$\neg(\alpha \vee \beta)$	$\equiv$	$(\neg\alpha \wedge \neg\beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma))$	$\equiv$	$((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma))$	$\equiv$	$((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

$(p \Rightarrow \neg(q \wedge r$

$\neg p \vee \neg(q \wedge r$

Can be used to rewrite formulas….

$(p \Rightarrow \neg(q \wedge r))$ $\quad$ $\neg(q \wedge r) \vee \neg p$

$(q \wedge r) \Rightarrow \neg p$ $\quad$ $\neg q \vee \neg r \vee \neg p$

Validity and satisfiability

A sentence is valid if it is true in **all** interpretations

e.g., $True$, $A \lor \neg A$, $A \Rightarrow A$, $(A \land (A \Rightarrow B)) \Rightarrow B$

Validity is connected to inference via the Deduction Theorem:

$KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid

A sentence is satisfiable if it is true in **some** interpretation

e.g., $A \lor B$, C

A sentence is unsatisfiable if it is true in **no** interpretations

e.g., $A \land \neg A$

Satisfiability is connected to inference via the following:

$KB \models \alpha$ if and only if $(KB \land \neg\alpha)$ is unsatisfiable

i.e., prove α by *reductio ad absurdum*

Validity and Satisfiability

$\langle \alpha$ is valid iff $\neg\alpha$ unsatisfiable $\rangle$

$\langle \alpha$ is satisfiable iff $\neg\alpha$ is valid $\rangle$

The statements above are:

A: All false

B: Some true Some false

C: All true

Validity and Satisfiability

iclicker.

cannot be true in any model

true in all models

⟨α is valid iff ¬α unsatisfiable⟩ T

F

⟨α is satisfiable iff ¬α is valid⟩ F

true in some models — true in all models

The statements above are:

A: All false

B: Some true Some false

C: All true

Lecture Overview

- Basics Recap: Interpretation / Model /
- Propositional Logics
- Satisfiability, Validity
- **Resolution in Propositional logics**

Proof by resolution

Key ideas

proof

$KB \models \alpha$

show

equivalent to : $KB \wedge \neg\alpha$ *unsatifiable*

- Simple Representation for Conjunctive Normal Form
- Simple Rule of Derivation

Resolution

Conjunctive Normal Form (CNF)

Rewrite $KB \wedge \neg\alpha$ into **conjunction of disjunctions**

literals

$$\underbrace{(A \vee \neg B)}_{\text{Clause}} \wedge \underbrace{(B \vee \neg C \vee \neg D)}_{\text{Clause}}$$

- Any KB can be converted into CNF !

Example: Conversion to CNF

$A \Leftrightarrow (B \vee C)$

1. Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.
   $(A \Rightarrow (B \vee C)) \wedge ((B \vee C) \Rightarrow A)$

2. Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.
   $(\neg A \vee B \vee C) \wedge (\neg(B \vee C) \vee A)$

3. Using de Morgan's rule replace $\neg(\alpha \vee \beta)$ with $(\neg\alpha \wedge \neg \beta)$:
   $(\neg A \vee B \vee C) \wedge ((\neg B \wedge \neg C) \vee A)$

4. Apply distributive law ($\vee$ over $\wedge$) and flatten:
   $(\neg A \vee B \vee C) \wedge (\neg B \vee A) \wedge (\neg C \vee A)$

Example: Conversion to CNF

$A \Leftrightarrow (B \lor C)$

5. KB is the conjunction of all of its sentences (all are true), so write each clause (disjunct) as a sentence in KB:

…
$(\neg A \lor B \lor C)$
$(\neg B \lor A)$
$(\neg C \lor A)$
…

Resolution Deduction step

Resolution: inference rule for CNF: sound and complete! *

$(A \vee B \vee C)$

$(\neg A)$

$-------------$

$\therefore (B \vee C)$

"If A or B or C is true, but not A, then B or C must be true."

$(A \vee B \vee C)$

$(\neg A \vee D \vee E)$

$------------$

$\therefore (B \vee C \vee D \vee E)$

"If A is false then B or C must be true, or if A is true then D or E must be true, hence since A is either true or false, B or C or D or E must be true."

$(A \vee B)$

$(\neg A \vee B)$

$---------$

$\therefore (B \vee B) \equiv B$

Simplification

Learning Goals for today's class

You can:

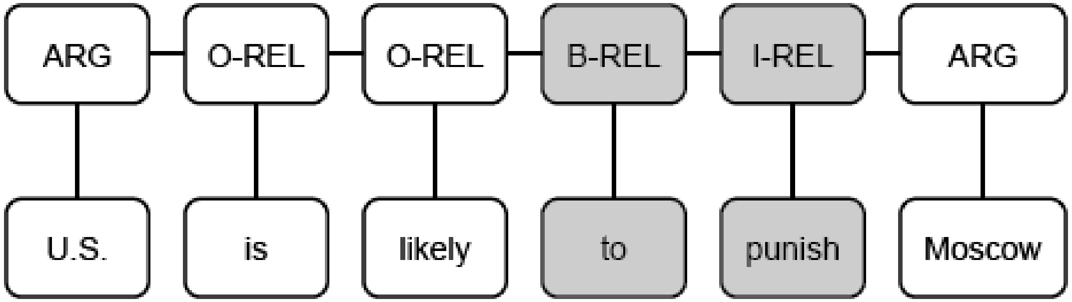
- Describe relationships between different logics
- Apply the definitions of Interpretation, model, logical entailment, soundness and completeness
- Define and apply satisfiability and validity
- Convert any formula to CNF
- Justify and apply the resolution step

PhD thesis I was reviewing some months ago…
University of Alberta
EXTRACTING INFORMATION NETWORKS FROM TEXT

We model *predicate detection* as a sequence labeling problem — …. We adopt the BIO encoding, a widely-used technique in NLP.

Our method, called Meta-CRF, is based on Conditional Random Fields (CRF) .

CRF is a graphical model that estimates a conditional probability distribution, denoted p(yjx), over label sequence y given the token sequence x.



Next class Fri

- Finish Resolution

- Another proof method for Prop. Logic
  Model checking - Searching through truth assignments. Walksat.

- First Order Logics

Ignore from this slide forward

Preview: we will define a logic (first-order logic) which is expressive enough to say almost anything of interest, and for which there exists a sound and complete inference procedure.

That is, the procedure will answer any question whose answer follows from what is known by the KB.

Try it Yourselves

- 7.9 page 238: (Adapted from Barwise and Etchemendy (1993).) If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.

- *Derive the KB in normal form.*
- *Prove: Horned, Prove: Magical.*

Exposes useful constraints

- **"You can't learn what you can't represent."** --- G. Sussman
- **In logic:** *If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.*

  Prove that the unicorn is both magical and horned.
- **A good representation makes this problem easy:**

$$(\neg Y \vee \neg R) \wedge (Y \vee R) \wedge (Y \vee M) \wedge (R \vee H) \wedge (\neg M \vee H) \wedge (\neg H \vee G)$$

Resolution

Conjunctive Normal Form (CNF—universal)

conjunction of $\underbrace{\textbf{disjunctions} \text{ of } \textbf{literals}}_{\textbf{clauses}}$

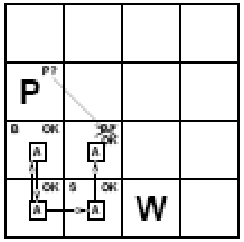
E.g., $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

Resolution inference rule (for CNF): complete for propositional logic

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m_1 \vee \cdots \vee m_n}{\ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

where ℓ_i and m_j are complementary literals. E.g.,

$$\frac{P_{1,3} \vee P_{2,2}, \qquad \neg P_{2,2}}{P_{1,3}}$$



Resolution is sound and complete for propositional logic

Conversion to CNF

$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

1. Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.

   $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

2. Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.

   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$

3. Move $\neg$ inwards using de Morgan's rules and double-negation:

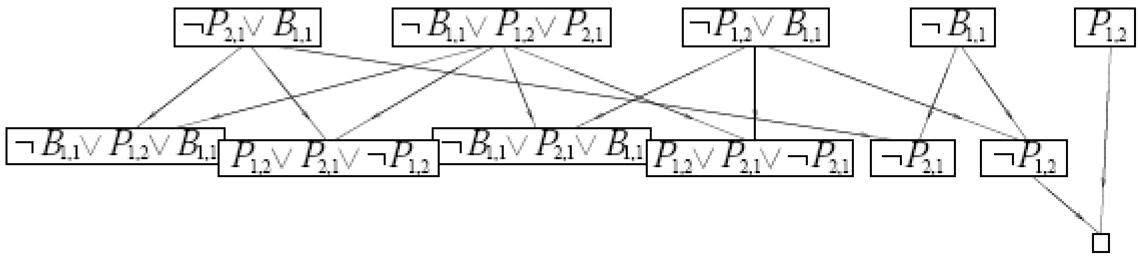
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$

4. Apply distributivity law ($\vee$ over $\wedge$) and flatten:

   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

Resolution example

$KB = (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$ $\alpha = \neg P_{1,2}$



Forward, backward chaining are linear-time, complete for Horn clauses
Resolution is complete for propositional logic

Propositional logic lacks expressive power

Logical equivalence

To manipulate logical sentences we need some rewrite rules.

Two sentences are logically equivalent iff they are true in same models: α ≡ ß iff α ╞ β and β ╞ α

$$
\begin{aligned}
(\alpha \land \beta) &\equiv (\beta \land \alpha) && \text{commutativity of } \land \\
(\alpha \lor \beta) &\equiv (\beta \lor \alpha) && \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) &\equiv (\alpha \land (\beta \land \gamma)) && \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) &\equiv (\alpha \lor (\beta \lor \gamma)) && \text{associativity of } \lor \\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \lor \beta) && \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \land \beta) &\equiv (\neg\alpha \lor \neg\beta) && \text{de Morgan} \\
\neg(\alpha \lor \beta) &\equiv (\neg\alpha \land \neg\beta) && \text{de Morgan} \\
(\alpha \land (\beta \lor \gamma)) &\equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) && \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) &\equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) && \text{distributivity of } \lor \text{ over } \land
\end{aligned}
$$

Validity and satisfiability

A sentence is valid if it is true in all models,

e.g., *True*, $A \vee \neg A$, $A \Rightarrow A$, $(A \wedge (A \Rightarrow B)) \Rightarrow B$

(tautologies)

Validity is connected to inference via the Deduction Theorem:

$KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid

A sentence is satisfiable if it is true in some model

e.g., $A \vee B$, C

(determining satisfiability of sentences is NP-complete)

A sentence is unsatisfiable if it is false in all

Proof methods

Proof methods divide into (roughly) two kinds:

Application of inference rules:

Legitimate (sound) generation of new sentences from old.

- ✓ Resolution
- ✓ Forward & Backward chaining

Model checking

Searching through truth assignments.

- ✓ Improved backtracking: Davis--Putnam-Logemann-Loveland (DPLL)
- ✓ Heuristic search in model space: Walksat.

Normal Form

We want to prove:

$$KB \models \alpha$$

$$\text{equivalent to}: KB \wedge \neg\alpha \text{ unsatifiable}$$

We first rewrite $KB \wedge \neg\alpha$ into conjunctive normal form (CNF).

A "conjunction of disjunctions"

literals

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

Clause Clause

- Any KB can be converted into CNF
- k-CNF: exactly k literals per clause

Example: Conversion to CNF

$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

1. Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.
   $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

2. Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$

3. Move $\neg$ inwards using de Morgan's rules and double-negation:
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$

4. Apply distributive law ($\wedge$ over $\vee$) and flatten:
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

Resolution Inference Rule for CNF

$(A \vee B \vee C)$

$(\neg A)$

$-----------$

$\therefore (B \vee C)$

"If A or B or C is true, but not A, then B or C must be true."

$(A \vee B \vee C)$

$(\neg A \vee D \vee E)$

$-----------$

$\therefore (B \vee C \vee D \vee E)$

"If A is false then B or C must be true,
or if A is true then D or E must be true,
hence since A is either true or false, B or C or D or E must be true."

$(A \vee B)$

$(\neg A \vee B)$

$--------$ ← Simplification

$\therefore (B \vee B) \equiv B$

Resolution Algorithm

- The resolution algorithm tries to prove: $KB \models \alpha$ *equivalent to* $KB \wedge \neg\alpha$ *unsatisfiable*

- Generate all new sentences from KB and the query.
- One of two things can happen:

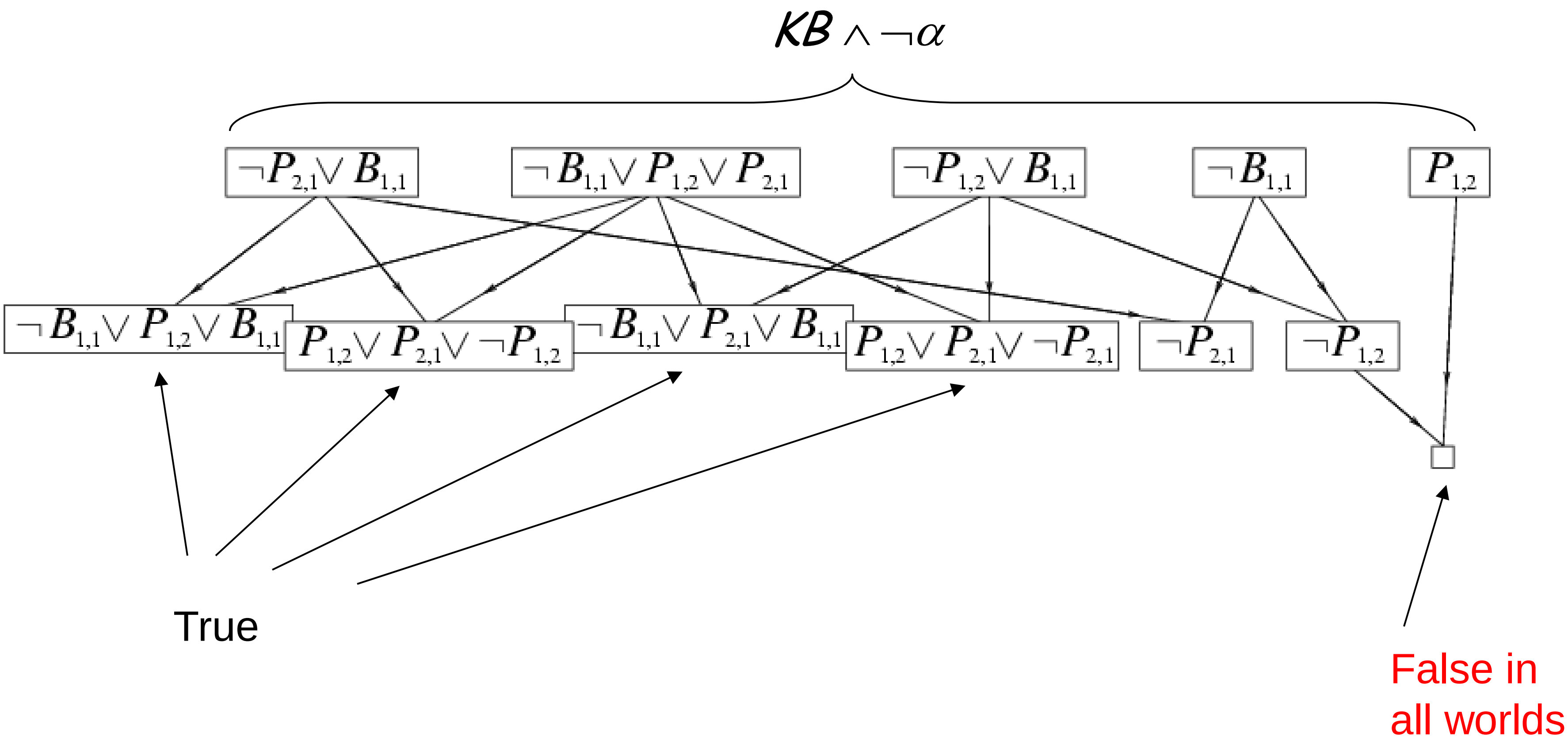
1. We find $P \wedge \neg P$ which is unsatisfiable, i.e. we can entail the query.

2. We find no contradiction: there is a model that satisfies the Sentence (non-trivial) and hence we cannot entail the query.

$$KB \wedge \neg\alpha$$

Resolution example

- $KB = (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$
- $\alpha = \neg P_{1,2}$



Horn Clauses

- Resolution in general can be exponential in space and time.

- If we can reduce all clauses to "Horn clauses" resolution is linear in space and time

A clause with at most 1 positive literal.
e.g. $A \vee \neg B \vee \neg C$

- Every Horn clause can be rewritten as an implication with a conjunction of positive literals in the premises and a single positive literal as a conclusion.

e.g. $B \wedge C \Rightarrow A$

- 1 positive literal: definite clause

- 0 positive literals: Fact or integrity constraint:
  e.g. $(\neg A \vee \neg B) \equiv (A \wedge B \Rightarrow False)$

Normal Form

We want to prove: $KB \models \alpha$

$\textit{equivalent to}: KB \wedge \neg\alpha \ \textit{unsatifiable}$

We first rewrite $KB \wedge \neg\alpha$ into conjunctive normal form (C

A "conjunction of disjunctions"

literals

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

Clause Clause

- Any KB can be converted into CNF
- k-CNF: exactly k literals per clause

Example: Conversion to CNF

$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

1. Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.
   $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

2. Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$

3. Move $\neg$ inwards using de Morgan's rules and double-negation:
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$

4. Apply distributive law ($\wedge$ over $\vee$) and flatten:
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

Resolution Inference Rule for CNF

$(A \vee B \vee C)$

$(\neg A)$

$\therefore (B \vee C)$

"If A or B or C is true, but not A, then B or C must be true."

$(A \vee B \vee C)$

$(\neg A \vee D \vee E)$

$\therefore (B \vee C \vee D \vee E)$

"If A is false then B or C must be true,
or if A is true then D or E must be true, hence since A is either true or false, B or C or D or E must be true."

$(A \vee B)$

$(\neg A \vee B)$

$\therefore (B \vee B) \equiv B$

Simplification

Resolution Algorithm

- The resolution algorithm tries to prove: $KB \models \alpha$ *equivalent to* $KB \wedge \neg\alpha$ *unsatisfiable*

- Generate all new sentences from KB and the query.
- One of two things can happen:

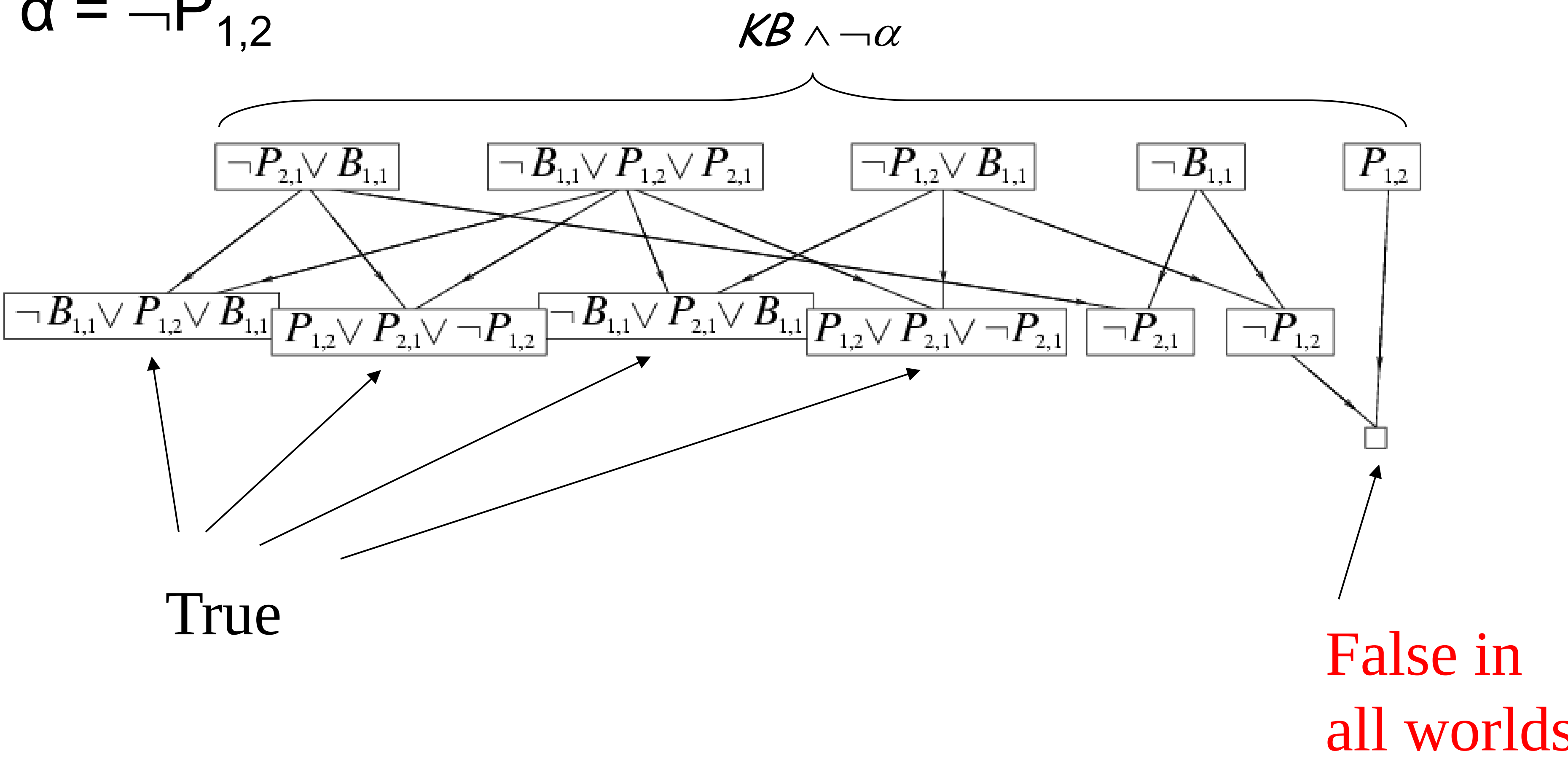
1. We find $P \wedge \neg P$ which is unsatisfiable, i.e. we can entail the query.

2. We find no contradiction: there is a model that satisfies the Sentence (non-trivial) and hence we cannot entail the query. $KB \wedge \neg\alpha$

Resolution example

$KB = (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$

$\alpha = \neg P_{1,2}$



Horn Clauses

- Resolution in general can be exponential in space and time.

- If we can reduce all clauses to "Horn clauses" resolution is li

  A clause with at most 1 positive literal.

  e.g. $A \vee \neg B \vee \neg C$

  - Every Horn clause can be rewritten as an implication with a conjunction of positive literals in the premises and a sin positive literal as a conclusion.

  e.g. $B \wedge C \Rightarrow A$

  $$(\neg A \vee \neg B) \equiv (A \wedge B \Rightarrow False)$$

  - 1 positive literal: definite clause