Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

Key Points

- Normal forms: definitions, know how to convert, applying basic laws and inference rules
- Theorem proving: basic approaches. forward and backward chaining concept, and resolution.
- know how to do resolution in propositional logic
- limitation of propositional logic

midterm exam range: up till here.

Slides03-logic.

Until yeah

You are screen sharing

Stop Share

Key Points

- Normal forms: definitions, know how to convert, applying basic laws and inference rules
- Theorem proving: basic approaches. forward and backward chaining concept, and resolution.
- know how to do resolution in propositional logic
- limitation of propositional logic

midterm
exam range: up till here.

slide03-logic.
p47

Until yeah, he's 47 not 97 my poor hand

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

F.O.L

Unification Algorithm

① Nat. Lang

→ F.O.L

② standard form

ⓐ prenex

ⓑ CNF

ⓒ skolem

③ Resolution

ⓐ Unification

$\neg P(x)$ vs $P(f(y))$

Let $W = \{E_1, \ldots, E_n\}$ be the set of expressions to be unified.

1. If necessary, **rename** variables so that no pair (E_i, E_j) from different clauses has any variables in common.
2. **Set** $k = 0, W_k = W, \sigma_k = \epsilon$ (empty substitution).
3. **If** W_k **is a singleton** (contains only one expr), stop; σ_k is a most general unifier for W. **Otherwise**, let D_k be the disagreement set for W_k.
4. **If** there exist elements v_k and t_k in D_k such that v_k **is a variable that does not occur in term** t_k, go to step 5. **Otherwise**, stop; W is not unifiable.
5. Let $\sigma_{k+1} = \sigma_k \circ \{v_k/t_k\}$ and $W_{k+1} = W_k\{v_k/t_k\}$. (Note that $W_{k+1} = W_k\sigma_{k+1}$)
6. Set $k = k + 1$ and go to step 3. ← Substitution

ⓑ Resolution step.

ⓒ algorithm.

So we are done up to that that point and we're going to look at the resolution step and actual algorithm

You are screen sharing Stop Share

$D = \{x, y\}$

Case 1: $v = x$, $t = y$
Case 2: $v = y$, $t = x$
both are ok

Unification Example

Step 1

$W = \{$ $P(x, f(x), z)$ vs.
$P(g(y), f(g(a)), y)$: $\}$

① $D_1 = \{x, g(y)\}$
② $v = x$, $t = g(y)$
$\cancel{v = g(y), t = x}$
$\sigma = \{x/g(y)\}$
③ $\{\ldots\}\sigma$
$W\sigma$

1. $\{x/g(y)\}$:
   $P(g(y), f(g(y)), z)$
   $P(g(y), f(g(a)), y)$

Step 2

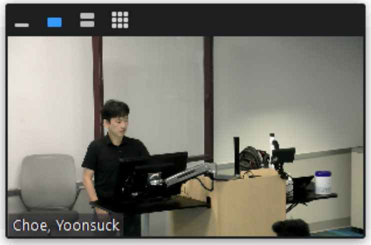
$= W$ ① $D_2 = \{y, a\}$ (var, const)
② $v = y$, $t = a$
$\sigma \circ \{y/a\} \rightarrow \sigma$.
③ $W\sigma$

2. $\{y/a\}$:
   $P(g(a), f(g(a)), z)$
   $P(g(a), f(g(a)), a)$

3. $\{z/a\}$:
   $P(g(a), f(g(a)), a)$

Unifier: $\{x/g(a), y/a, z/a\}$ — σ

You apply it, apply the Sigma to W.



$D = \{x, y\}$

Case 1: $v = x$, $t = y$
Case 2: $v = y$, $t = x$
both are ok

Unification Example

Step 1

$W = \{ P(x, f(x), z)$ vs. $P(g(y), f(g(a)), y) \}$:

1. $\{x/g(y)\}$:
   $P(g(y), f(g(y)), z)$
   $P(g(y), f(g(a)), y)$ $= W$
2. $\{y/a\}$:
   $P(g(a), f(g(a)), z)$
   $P(g(a), f(g(a)), a)$
3. $\{z/a\}$:
   $P(g(a), f(g(a)), a)$

Unifier: $\{x/g(a), y/a, z/a\}$ — σ

① $D_1 = \{x, g(y)\}$
② $v = x$, $t = g(y)$
~~$v = g(y)$, $t = x$~~
$\sigma = \{x/g(y)\}$
③ $\{ \dots \}\sigma$
$W\sigma$

Step 2

① $D_2 = \{y, a\}$ (var, const)
② $v = y$, $t = a$
$\sigma \circ \{y/a\} \to \sigma$.
③ $W\sigma$

Step

So now you, when, when when finally you reach step 3

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

$D = \{x, y\}$

Case1: $v = x$, $t = y$

Case2: $v = y$, $t = x$

both are ok

Unification Example

$W = \{ P(x, f(x), z)$ vs. $P(g(y), f(g(a)), y) \}$:

1. $\{x/g(y)\}$:

   $P(g(y), f(g(y)), z)$

   $P(g(y), f(g(a)), y)$ $= W$

2. $\{y/a\}$:

   $P(g(a), f(g(a)), z)$

   $P(g(a), f(g(a)), a)$

3. $\{z/a\}$:

   $P(g(a), f(g(a)), a)$

Unifier: $\{x/g(a), y/a, z/a\}$

σ

Step 1

① $D_1 = \{x, g(y)\}$

② $v = x$, $t = g(y)$

$v = \cancel{g(y)}, \cancel{t = x}$

$\sigma = \{x/g(y)\}$

③ $\{\ldots\}\ \sigma$

$W\sigma$

Step 2

var ... const

① $D_2 = \{y, a\}$

② $v = y$, $t = a$

$\sigma\ \circ \{y/a\} \rightarrow \sigma$

③ $W\sigma \rightarrow W$

Step3 var

① $D_3 = \{z, a\}$

const

② $v = z$, $t = a$

$\sigma \circ \{z/a\} \rightarrow \sigma$

So when you apply that, and you apply this to

You are screen sharing Stop Share

$D = \{x, y\}$

Case 1: $v = x$, $t = y$
Case 2: $v = y$, $t = x$
both are ok

Unification Example

Step 1

① $D_1 = \{x, g(y)\}$

② $v = x$, $t = g(y)$

$v = g(y)$, ~~$t = x$~~

$\sigma = \{x/g(y)\}$

③ $\{\ldots\}\sigma$

$W\sigma$

$W = \{ P(x, f(x), z)$ vs. $P(g(y), f(g(a)), y)$: $\}$

1. $\{x/g(y)\}$:
$P(g(y), f(g(y)), z)$
$P(g(y), f(g(a)), y)$

2. $\{y/a\}$:
$P(g(a), f(g(a)), z)$
$P(g(a), f(g(a)), a)$

3. $\{z/a\}$:
$P(g(a), f(g(a)), a)$

Unifier: $\{x/g(a), y/a, z/a\}$

σ

① ∘ ② ∘ ③ ⇒ unifier

Step 2

$\sigma = W$

① $D_2 = \{y, a\}$ (var, const)

② $v = y$, $t = a$

$\sigma \circ \{y/a\} \rightarrow \sigma$

③ $W\sigma \rightarrow W$

W

Step 3

① $D_3 = \{z, a\}$ (var, const)

② $v = z$, $t = a$

$\sigma \circ \{z/a\} \rightarrow \sigma$

So this would be a unifier

Unification Example

g(a) g(a) a

$P(\cancel{x}, f(\cancel{x}), \cancel{z})$ vs.

$P(g(\cancel{y}), f(g(a)), \cancel{y})$:

a a

1. $\{x/g(y)\}$:
   $P(g(y), f(g(y)), z)$
   $P(g(y), f(g(a)), y)$

2. $\{y/a\}$:
   $P(g(a), f(g(a)), z)$
   $P(g(a), f(g(a)), a)$

3. $\{z/a\}$:
   $P(g(a), f(g(a)), a)$

Unifier: $\{x/g(a), y/a, z/a\}$

Unification Example

$g(a)$ $\quad$ $g(a)$

$P(\cancel{x}, f(\cancel{x}), \cancel{z})$ vs.

$P(g(\cancel{y}), f(g(a)), \cancel{y})$:

a $\quad$ a

1. $\{x/g(y)\}$:
   $P(g(y), f(g(y)), z)$
   $P(g(y), f(g(a)), y)$

2. $\{y/a\}$:
   $P(g(a), f(g(a)), z)$
   $P(g(a), f(g(a)), a)$

3. $\{z/a\}$:
   $P(g(a), f(g(a)), a)$

Unifier: $\{x/g(a), y/a, z/a\}$

$$\frac{P(g(a), f(g(a)), a)}{P(g(a), f(g(a)), a)}$$

$=$

Unification Example

g(a) g(a) a

$P(\cancel{x}, f(\cancel{x}), \cancel{z})$ vs.

$P(g(\cancel{y}), f(g(a)), \cancel{y})$:

a a

$$P(g(a), f(g(a)), a) = P(g(a), f(g(a)), a)$$

1. $\{x/g(y)\}$:
   $P(g(y), f(g(y)), z)$
   $P(g(y), f(g(a)), y)$

2. $\{y/a\}$:
   $P(g(a), f(g(a)), z)$
   $P(g(a), f(g(a)), a)$

3. $\{z/a\}$:
   $P(g(a), f(g(a)), a)$

Unifier: $\{x/g(a), y/a, z/a\}$ ✓

Okay. So again, this

22 FALL CSCE 420 501: ARTIFICIA | Launch Meeting - Zoom

canvas.tamu.edu/courses/171555

22 FALL CSCE 420 5... | 22 FALL CSCE 633 6... | Video Conferencing...

Mouse | Select | Text | Draw | Stamp | Spotlight | Eraser | Format | Undo | Redo | Clear | Save

Who can see what you share here? Recording On

Account | Dashboard | Courses | Calendar | Inbox | History | Commons | Help

CS.Fall.2022

- Home
- Syllabus
- Pages
- Files
- Assignments
- Grades
- Accessibility Report
- Core Curriculum Assessment
- Course Evaluations
- Mediasite Collection
- Announcements
- Modules
- Discussions
- People
- Outcomes
- Rubrics
- Quizzes
- Collaborations
- Settings

2	8/30/2022	9/1/2022	Search (uninformed)	Ch 3.1-3.6	Ch 3.1-3.6	hw1 announced		slide02-search-and-gameplaying.pdf	8/30: last day for add/drop
3	9/6/2022	9/8/2022	Search (informed)	Ch 4.1-4.3	Ch 4.1-4.3			slide02-search-and-gameplaying.pdf	
4	9/13/2022	9/15/2022	Game playing	Ch 5	Ch 5			slide02-search-and-gameplaying.pdf	
5	9/20/2022	9/22/2022	Propositional logic and theorem proving	Ch 7	Ch 7	hw2 announced	hw1 due 9/25 11:59pm	slide03-logic	
6	9/27/2022	9/29/2022	First-order logic and theorem proving	Ch 8, 9	Ch 8, 9			slide03-logic	
7	10/4/2022	10/6/2022	First-order logic and theorem proving; Prolog ; Planning;	Ch 8, 9	Ch 8, 9			slide03-logic, slide04-planning	
8	no class	10/13/2022 **Midterm Exam**	**Midterm Exam (Thu, in class)**	Ch 10	Ch 11	hw3 announced	hw2 due: 10/11 Tuesday 11:59pm!!		10/10: midsemester grades due
9	10/18/2022	10/20/2022	Uncertainty and probabilistic reasoning	Ch 13, 14	Ch 12, 13				
10	10/25/2022	10/27/2022	Uncertainty and probabilistic reasoning	Ch 13, 14	Ch 12, 13				
11	11/1/2022	11/3/2022	Machine learning	Ch 18.1-	Ch 19.1-	hw			

Today

next Thu: midterm exam!

HW#1 Solution TBA: this week.

Fall Break

Looking forward to the full break. Good

You are screen sharing | Stop Share

Type here to search

11:10 AM 10/4/2022

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

Unifier

$\sigma = \{x/f(z)\} \rightarrow p(f(z))$

$\neg p(f(z)) \rightarrow$ False.

KB

C_1 $p(x)$

C_2

$\vdots$

C_n $\neg p(f(z))$

C_{n+1}

$\neg G$ C_{n+2}

C_{n+3}

$\vdots$

False

Resolution in Predicate Calculus

- Factors
- Binary resolvent
- Properties of resolution

Oh, I guess a little bit more complicated than this, but usually this is

You are screen sharing Stop Share

$C_1: P(x) \vee P(a) \vee Q(x)$

$C_2: \neg P(y)$

Factor

Unify $P(x)$, $P(a) \rightarrow P(a)$

$\sigma = \{x/a\}$

Factor of a Clause

Definition: If two or more literals of a clause C (with the **same** sign) have a most general unifier σ, then $C\sigma$ is called a **Factor** of C. If $C\sigma$ is a unit clause, it is called a **Unit Factor** of C.

Example: $C = P(x) \vee P(f(y)) \vee \neg Q(x)$.

- The first two literals have a unifier $\sigma = \{x/f(y)\}$, so C has a factor $C\sigma = P(f(y)) \vee \neg Q(f(y))$.

Note: Factors of a clause are much succinct and when two clauses C_1 and C_2 cannot be resolved directly, their factors (let's call them C_1' and C_2' **can be** resolved.

C_1: $P(\cancel{x}) \vee P(a) \vee Q(\cancel{x})$

C_2: $\neg P(y)$

Factors

Unify $P(x)$, $P(a)$ → $P(a)$

$\sigma = \{x/a\}$

Factor of a Clause

Definition: If two or more literals of a clause C (with the **same** sign) have a most general unifier σ, then $C\sigma$ is called a **Factor** of C. If $C\sigma$ is a unit clause, it is called a **Unit Factor** of C.

Example: $C = P(x) \vee P(f(y)) \vee \neg Q(x)$.

- The first two literals have a unifier $\sigma = \{x/f(y)\}$, so C has a factor $C\sigma = P(f(y)) \vee \neg Q(f(y))$.

Note: Factors of a clause are much succinct and when two clauses C_1 and C_2 cannot be resolved directly, their factors (let's call them C_1' and C_2' **can be** resolved.

Factor of C_1: $P(a) \vee Q(a)$

C_2: $\neg P(y)$

a

$Q(a)$

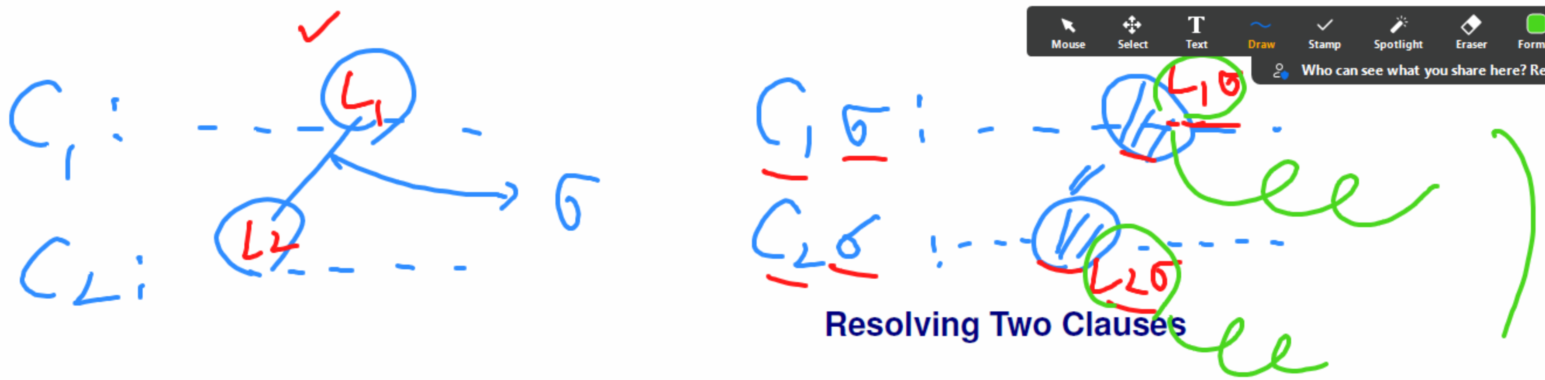
Factor of a Clause

Definition: If two or more literals of a clause C (with the **same** sign) have a most general unifier σ, then $C\sigma$ is called a **Factor** of C. If $C\sigma$ is a unit clause, it is called a **Unit Factor** of C.

Example: $C = P(x) \vee P(f(y)) \vee \neg Q(x)$.

- The first two literals have a unifier $\sigma = \{x/f(y)\}$, so C has a factor $C\sigma = P(f(y)) \vee \neg Q(f(y))$.

Note: Factors of a clause are much succinct and when two clauses C_1 and C_2 cannot be resolved directly, their factors (let's call them C_1' and C_2' **can be** resolved.



Resolving Two Clauses

Definition: Let C_1 and C_2 be two clauses (called *parent clauses*) with no variables in common, and with complementary literals L_1 and L_2 such that L_1 and $\neg L_2$ have a most general unifier σ. Then the clause

$$(C_1\sigma - L_1\sigma) \cup (C_2\sigma - L_2\sigma)$$

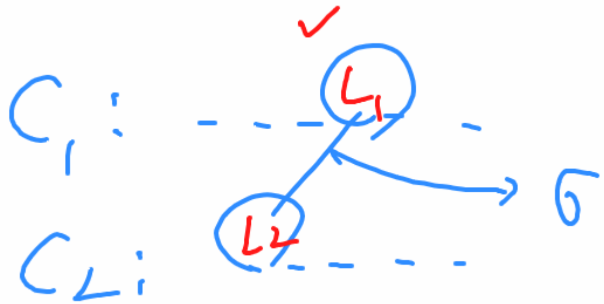
is called a **binary resolvent** of C_1 and C_2. The literals L_1 and L_2 are called **the literals resolved upon**.

Note: A clause can be treated as a set of literals.

$\{P(x)\} \cup \{Q(x)\} = \{P(x), Q(x)\} = P(x) \vee Q(x)$

Example: Resolve the following (hint: $\sigma = \{x/a\}$)

$C_1 = P(x) \vee Q(x)$ and $C_2 = \neg P(a) \vee R(y)$.



Resolving Two Clauses

$C_1 : \{ P(x), Q(x) \}$

$C_2 : \{ \neg P(a), R(y) \}$

Definition: Let C_1 and C_2 be two clauses (called *parent clauses*) with no variables in common, and with complementary literals L_1 and L_2 such that L_1 and $\neg L_2$ have a most general unifier σ. Then the clause

$$(C_1\sigma - L_1\sigma) \cup (C_2\sigma - L_2\sigma)$$

is called a **binary resolvent** of C_1 and C_2. The literals L_1 and L_2 are called **the literals resolved upon**.

Note: A clause can be treated as a set of literals.

$\{P(x)\} \cup \{Q(x)\} = \{P(x), Q(x)\} = P(x) \vee Q(x)$

Example: Resolve the following (hint: $\sigma = \{x/a\}$)

$C_1 = P(x) \vee Q(x)$ and $C_2 = \neg P(a) \vee R(y)$.

Resolving Two Clauses: Example Revisited

Example: Resolve the following (hint: $\sigma = \{x/a\}$)

$C_1 = P(x) \vee Q(x)$ and $C_2 = \neg P(a) \vee R(y)$

$L_1 \sigma$ $\qquad$ $L_2 \sigma$

$C_1 = P(x) \vee Q(x)$ $\qquad$ $C_2 = \neg P(a) \vee R(y)$

$\sigma = \{x/a\}$ $\qquad$ $\sigma = \{x/a\}$

$\downarrow$ $\qquad$ $\downarrow$

$C_1\sigma = \cancel{P(a)} \vee Q(a)$ $\qquad$ $C_2\sigma = \neg \cancel{P(a)} \vee R(y)$

remove $L_1\sigma = P(a)$ $\qquad$ remove $L_2\sigma = \neg P(a)$

$\downarrow$ $\qquad$ $\downarrow$

$(C_1\sigma - L_1\sigma) = Q(a)$ $\qquad$ $(C_2\sigma - L_2\sigma) = R(y)$

$\cup$

$P(x)$, $P(a)$ $\xrightarrow{\sigma}$ $\{x/a\}$

$\{Q(a), R(y)\}$

$=$

$Q(a) \vee R(y)$.

Resolvent

Definition: A *resolvent* of parent clauses C_1 and C_2 is one of the following binary resolvents:

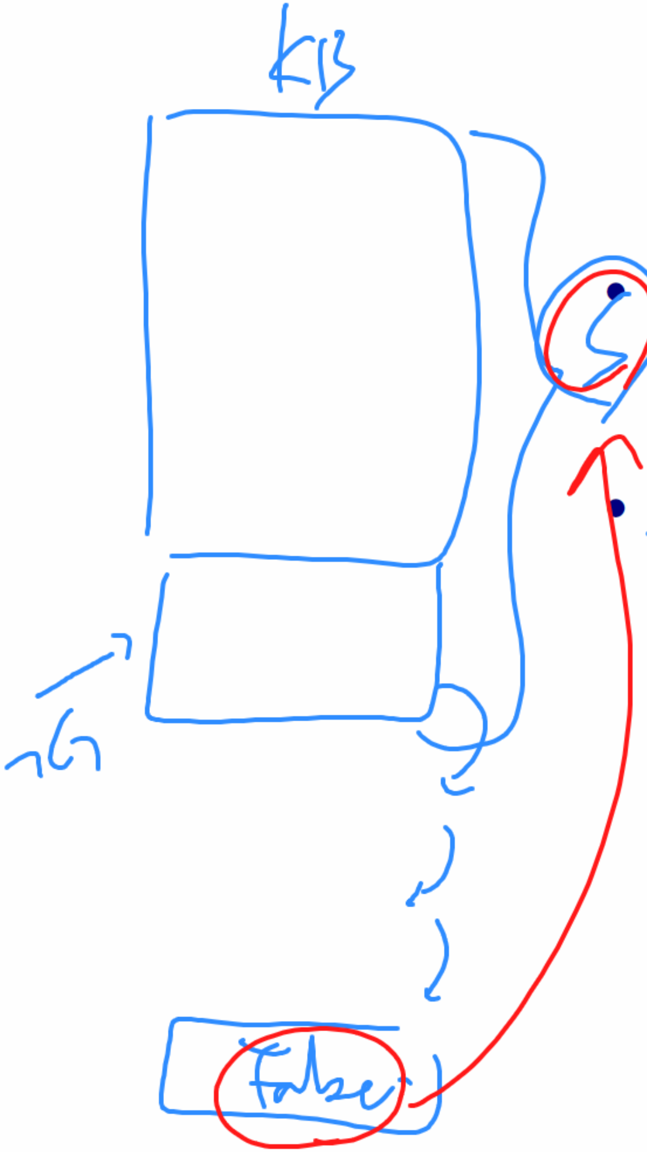
1. a binary resolvent of C_1 and C_2
2. a binary resolvent of C_1 and a factor of C_2
3. a binary resolvent of a factor of C_1 and C_2
4. a binary resolvent of a factor of C_1 and a factor of C_2

Example: resolve the two clauses

1. $C_1 = P(x) \vee P(f(y)) \vee R(g(y))$ and
2. $C_2 = \neg P(f(g(a))) \vee Q(b)$.

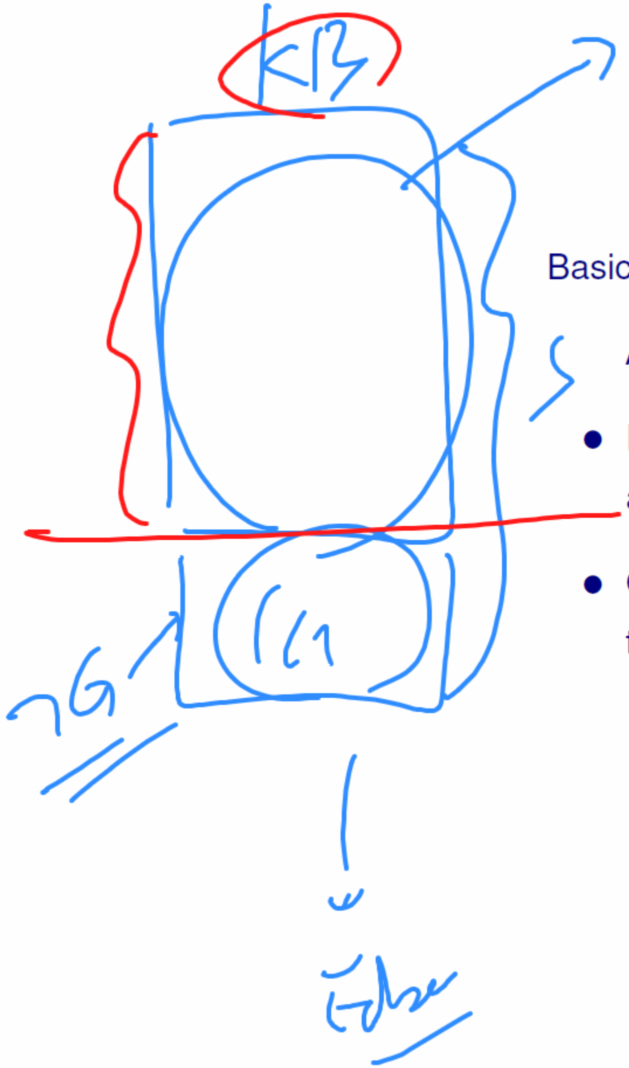
(hint: resolve the factor of C_1 and clause C_2)

Well factor of both. if possible, you can resolve. you can try



Property of Resolution for First-Order Logic

- **Complete**: If a set of clauses S is unsatisfiable, resolution will *eventually* derive $\mathbf{False}$.
  - *Everything that is true can be proved (eventually).*
- **Sound**: If $\mathbf{False}$ is derived by resolution, then the original set of clauses S is unsatisfiable.
  - *Everything that is proved is true.*



False

Weakness of Resolution

Basically, resolution tries to derive

Axioms $\wedge$ $\neg$ Theorem = **False**

- Is there a **False** in the axioms? If there is, the whole formula will always be unsatisfiable no matter what.
- Can we tell whether axioms alone can derive **False** ? (generally, this is not the case)

22 FALL CSCE 420 501: ARTIFICIA | Launch Meeting - Zoom

canvas.tamu.edu/courses/171555

22 FALL CSCE 420 5... | 22 FALL CSCE 633 6... | Video Conferencing...

Mouse | Select | Text | Draw | Stamp | Spotlight | Eraser | Format | Undo | Redo | Clear | Save

Who can see what you share here? Recording On

Account | Dashboard | Courses | Calendar | Inbox 7 | History | Commons | Help 10

CS.Fall.2022

- Home
- Syllabus
- Pages
- Files
- Assignments
- Grades
- Accessibility Report
- Core Curriculum Assessment
- Course Evaluations
- Mediasite Collection
- Announcements
- Modules
- Discussions
- People
- Outcomes
- Rubrics
- Quizzes
- Collaborations
- Settings

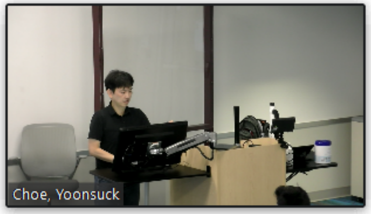
2	8/30/2022	9/1/2022	Search (uninformed)	Ch 3.1-3.6	Ch 3.1-3.6	hw1 announced		slide02-search-and-gameplaying.pdf	8/30: last day for add/drop
3	9/6/2022	9/8/2022	Search (informed)	Ch 4.1-4.3	Ch 4.1-4.3			slide02-search-and-gameplaying.pdf	
4	9/13/2022	9/15/2022	Game playing	Ch 5	Ch 5			slide02-search-and-gameplaying.pdf	
5	9/20/2022	9/22/2022	Propositional logic and theorem proving	Ch 7	Ch 7	hw2 announced	hw1 due 9/25 11:59pm	slide03-logic	
6	9/27/2022	9/29/2022	First-order logic and theorem proving	Ch 8, 9	Ch 8, 9			slide03-logic	
7	10/4/2022	10/6/2022	First-order logic and theorem proving; Prolog; Planning	Ch 8, 9	Ch 8, 9			slide03-logic, slide04-planning	
8	**no class**	10/13/2022 **Midterm Exam**	**Midterm Exam (Thu, in class)**	Ch 10	Ch 11	hw3 announced	hw2 due: 10/11 Tuesday 11:59pm!!		10/10: midsemester grades due
9	10/18/2022	10/20/2022	Uncertainty and probabilistic reasoning	Ch 13, 14	Ch 12, 13				
10	10/25/2022	10/27/2022	Uncertainty and probabilistic reasoning	Ch 13, 14	Ch 12, 13				
11	11/1/2022	11/3/2022	Machine learning	Ch 18.1-	Ch 19.1-	hw	hw3 due		

Today

next Thu: midterm exam!

HW#1 Solution TBA: this week.

Fall Break



So you can take a look at that and whom one solution will be announced, maybe today or tomorrow, so they can take a look at that

You are screen sharing | Stop Share

Type here to search

11:11 AM 10/4/2022

Resolvent: A Full Example

Example: resolve the two clauses

1. $C_1 = P(x) \lor P(f(y)) \lor R(g(y))$ and $\longrightarrow$
2. $C_2 = \neg P(f(g(a))) \lor Q(b)$.

Factor

$\sigma = \{x / f(\cdot)\}$

1. Get the factor of C_1:
$$C_1\{x/f(y)\} = \overset{C_1}{\boxed{P(f(y)) \lor R(g(y))}}$$
2. Resolve factor of C_1 and C_2:
$$P(f(y)) \lor R(g(y)) \text{ vs. } \neg P(f(g(a))) \lor Q(b)$$
3. $\sigma = \{y/g(a)\}$:
$$\underbrace{P(f(g(a)))}_{\text{remove}} \lor R(g(g(a))) \text{ vs. } \underbrace{\neg P(f(g(a)))}_{\text{remove}} \lor Q(b)$$
4. Result:
$$R(g(g(a))) \lor Q(b)$$

So the first close now become. See why, after factoring, finding the factor.

Resolvent: A Full Example

Factor

Example: resolve the two clauses

1. $C_1 = P(x) \vee P(f(y)) \vee R(g(y))$ and $\longrightarrow$ $\sigma = \{x/f(y)\}$
2. $C_2 = \neg P(f(g(a))) \vee Q(b) \leftarrow \sigma$

C_2

C_1

1. Get the factor of C_1:

$$C_1\{x/f(y)\} = P(f(y)) \vee R(g(y)) \leftarrow \sigma$$

$P(y)$

2. Resolve factor of C_1 and C_2:

$$P(f(y)) \vee R(g(y)) \text{ vs. } \neg P(f(g(a))) \vee Q(b)$$

$\neg P(f(g(a)))$

3. $\sigma = \{y/g(a)\}$:

$$\underbrace{P(f(g(a)))}_{\text{remove}} \vee R(g(g(a))) \text{ vs. } \underbrace{\neg P(f(g(a)))}_{\text{remove}} \vee Q(b)$$

$\sigma = \{y/f(g(a))\}$

4. Result:

$$R(g(g(a))) \vee Q(b)$$

Resolvent: A Full Example

Example: resolve the two clauses

1. $C_1 = P(x) \vee P(f(y)) \vee R(g(y))$ and
2. $C_2 = \neg P(f(g(a))) \vee Q(b)$.

Factor → $\sigma = \{x/f(y)\}$

C_2 $\leftarrow \sigma$

C_1

1. Get the factor of C_1:
$$C_1\{x/f(y)\} = P(f(y)) \vee R(g(y)) \leftarrow \sigma$$

$P(y)$

2. Resolve factor of C_1 and C_2:
$$P(f(y)) \vee R(g(y)) \text{ vs. } \neg P(f(g(a))) \vee Q(b)$$

$\neg P(f(g(a)))$

3. $\sigma = \{y/g(a)\}$:
$$\underbrace{P(f(g(a)))}_{\text{remove}} \vee R(g(g(a))) \text{ vs. } \underbrace{\neg P(f(g(a)))}_{\text{remove}} \vee Q(b)$$

$\sigma = \{y/f(g(a))\}$

4. Result:
$$R(g(g(a))) \vee Q(b)$$

This would be C. 2, the result of applying this uniform, and
this would be the result of applying the new fire to see one

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

Resolvent: A Full Example

Example: resolve the two clauses

1. $C_1 = P(x) \vee P(f(y)) \vee R(g(y))$ and
2. $C_2 = \neg P(f(g(a))) \vee Q(b)$. $\leftarrow \sigma$

C_2 C_1

Factor

$\sigma = \{x/f(y)\}$

1. Get the factor of C_1:
$$C_1\{x/f(y)\} = \boxed{P(f(y)) \vee R(g(y))} \leftarrow \sigma$$
2. Resolve factor of C_1 and C_2:
$$P(f(y)) \vee R(g(y)) \text{ vs. } \neg P(f(g(a))) \vee Q(b)$$
3. $\sigma = \{y/g(a)\}$:
$$\underbrace{\cancel{P(f(g(a)))}}_{\text{remove}} \vee R(g(g(a))) \text{ vs. } \underbrace{\cancel{\neg P(f(g(a)))}}_{\text{remove}} \vee Q(b)$$
4. Result:
$$R(g(g(a))) \vee Q(b)$$

$P(y)$

$\neg P(f(g(a)))$

$\sigma = \{y/f(g(a))\}$

P. of f of T. V. and then not give it for t of A. to get rid of that, and then combine these to finally get this

You are screen sharing Stop Share

Example Proof Using Resolution [a]

Given: (1) The customs officials searched everyone who entered the country who was not a VIP. (2) Some of the drug dealers entered the country, and they were only searched by drug dealers. (3) No drug dealer was a VIP.

Prove: (4) Some of the customs officials were drug dealers.

[a] Chang & Lee, Example 5.22

Example: Predicates

1. $C(x)$: x is a customs official
2. $E(x)$: x entered the country
3. $V(x)$: x is a VIP

   search.
4. $S(x, y)$: x was searched by y

   $S(y, x)$
5. $D(x)$: x is a drug dealer

Example: English to First Order Logic

(1) The customs officials searched everyone who entered the country who was not a VIP. (2) Some of the drug dealers entered the country, and they were only searched by drug dealers. (3) No drug dealer was a VIP. (4) Some of the customs officials were drug dealers.

1. $\forall x((E(x) \land \neg V(x)) \rightarrow \exists y(S(x, y) \land C(y)))$
2. $\exists x(E(x) \land D(x) \land \forall y(S(x, y) \rightarrow D(y)))$
3. $\forall x(D(x) \rightarrow \neg V(x))$
4. $\exists x(D(x) \land C(x))$

So this process ends here. So this it was quantifier affects all of these

Example: English to First Order Logic

(1) The customs officials searched everyone who entered the country who was not a VIP. (2) Some of the drug dealers entered the country, and they were only searched by drug dealers. (3) No drug dealer was a VIP. (4) Some of the customs officials were drug dealers.

1. $\forall x((E(x) \wedge \neg V(x)) \rightarrow \exists y(S(x,y) \wedge C(y)))$
2. $\exists x(E(x) \wedge D(x) \wedge \forall y(S(x,y) \rightarrow D(y)))$
3. $\forall x(D(x) \rightarrow \neg V(x))$
4. $\exists x(D(x) \wedge C(x))$

Okay, so

Example: English to First Order Logic

Pers

(1) The customs officials searched everyone who entered the country who was not a VIP. (2) Some of the drug dealers entered the country, and they were only searched by drug dealers. (3) No drug dealer was a VIP. (4) Some of the customs officials were drug dealers.

1. $\forall x(E(x) \land \neg V(x)) \rightarrow \exists y(S(x, y) \land C(y)))$
2. $\exists x(E(x) \land D(x) \land \forall y(S(x, y) \rightarrow D(y)))$
3. $\forall x(D(x) \rightarrow \neg V(x))$
4. $\exists x(D(x) \land C(x))$

So this person X was searched by someone. Why, who is a customs official? Okay, So that

Example: English to First Order Logic

(1) The customs officials searched everyone who entered the country who was not a VIP. (2) Some of the drug dealers entered the country, and they were only searched by drug dealers. (3) No drug dealer was a VIP. (4) Some of the customs officials were drug dealers.

1. $\forall x((E(x) \wedge \neg V(x)) \rightarrow \exists y(S(x,y) \wedge C(y)))$
2. $\exists x(E(x) \wedge D(x) \wedge \forall y(S(x,y) \rightarrow D(y)))$
3. $\forall x(D(x) \rightarrow \neg V(x))$
4. $\exists x(D(x) \wedge C(x))$

So 4 next source x by y implies drug deal y

22 FALL CSCE 420 501: ARTIFICIA | Launch Meeting - Zoom

canvas.tamu.edu/courses/171555

22 FALL CSCE 420 5... | 22 FALL CSCE 633 6... | Video Conferencing...

Mouse | Select | Text | Draw | Stamp | Spotlight | Eraser | Format | Undo | Redo | Clear | Save

Who can see what you share here? Recording On

Account
Dashboard
Courses
Calendar
Inbox
History
Commons
Help

CS.Fall.2022

- Home
- Syllabus
- Pages
- Files
- Assignments
- Grades
- Accessibility Report
- Core Curriculum Assessment
- Course Evaluations
- Mediasite Collection
- Announcements
- Modules
- Discussions
- People
- Outcomes
- Rubrics
- Quizzes
- Collaborations
- Settings

CSCE 420-501: Artificial Intelligence

Syllabus: [Syllabus]

Quick links and info:

- Office hours:
  - Instructor: TF 2:30pm-4:00pm [Zoom]
  - TA: Jin Hyun Park, jinhyun.park@tamu.edu ; MWF 9:30am-11:00am, [Zoom]
  - Grader: Vicente Balmaseda, vibalcam@tamu.edu (grading related questions only)
- Lecture notes: [Lecture notes folder]
- Lecture recordings: [Link]
- Practice Exams: [TBA]; Exam Rules [Link]

NEWS:

- [09/26] Homework 2 announced [Link]
  - IMPORTANT: Homework 2 is due on Tuesday! Right before the midterm exam.
  - Homework 2 does not include programming.
- [09/26]: slide04-planning uploaded [Lecture notes folder]
- [09/04]: Practice exams uploaded [Link]
  - Note: solutions will not be provided.
- [08/31]: Homework 1 announced [Link]
- [08/30]: Philosopher AI sample results [Link]
- [08/25]:
  - Lecture notes and screenshots: [Lecture notes folder]
  - Lecture recordings: [Link]

Weekly Schedule and Deadlines

Week	Tuesday	Thursday	Topic	Reading (3rd ed)	(4th ed)	Announcements	Dues	Lecture notes[Folder]	Comment
1	**no class**	8/25/2022	Introduction	Ch 1, 26.1, 26.2	Ch 1, 27.1, 27.2			slide01.pdf	

There are 2 things you wanna look at the practice exam actually well it's in the files. You can go

Import from Commons
Choose Home Page
View Course Stream
Course Setup Checklist
New Announcement
New Analytics
View Course Notifications

To Do

92 Grade Homework 1
100 points • Sep 25 at 11:59pm

2 Grade Homework 2
100 points • Oct 11 at 11:59pm

Coming Up View Calendar

Nothing for the next week

You are screen sharing | Stop Share

Type here to search

11:11 AM
10/4/2022

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

Choe, Yoonsuck

Example: English to First Order Logic

(1) The customs officials searched everyone who entered the country who was not a VIP. (2) Some of the drug dealers entered the country, and they were only searched by drug dealers. (3) No drug dealer was a VIP. (4) Some of the customs officials were drug dealers.

1. $\forall x((E(x) \wedge \neg V(x)) \rightarrow \exists y(S(x,y) \wedge C(y)))$
2. $\exists x(E(x) \wedge D(x) \wedge \forall y(S(x,y) \rightarrow D(y)))$
3. $\forall x(D(x) \rightarrow \neg V(x))$
4. $\exists x(D(x) \wedge C(x))$

So you have to think about different ways. to paraphrase it.

You are screen sharing Stop Share

Example: English to First Order Logic

(1) The customs officials searched everyone who entered the country who was not a VIP. (2) Some of the drug dealers entered the country, and they were only searched by drug dealers. (3) No drug dealer was a VIP. (4) Some of the customs officials were drug dealers.

1. $\forall x((E(x) \wedge \neg V(x)) \rightarrow \exists y(S(x,y) \wedge C(y)))$
2. $\exists x(E(x) \wedge D(x) \wedge \forall y(S(x,y) \rightarrow D(y)))$
3. $\forall x(D(x) \rightarrow \neg V(x))$
4. $\exists x(D(x) \wedge C(x))$

$\neg \exists x(D(x) \wedge V(x))$

Example: English to First Order Logic

(1) The customs officials searched everyone who entered the country who was not a VIP. (2) Some of the drug dealers entered the country, and they were only searched by drug dealers. (3) No drug dealer was a VIP. (4) Some of the customs officials were drug dealers.

1. $\forall x((E(x) \wedge \neg V(x)) \rightarrow \exists y(S(x,y) \wedge C(y)))$
2. $\exists x(E(x) \wedge D(x) \wedge \forall y(S(x,y) \rightarrow D(y)))$
3. $\forall x(D(x) \rightarrow \neg V(x))$
4. $\exists x(D(x) \wedge C(x))$

$\forall x \exists y$ $\{y/f(x)\}$

Example: Standard Form (I)

(1) $\forall x((E(x) \wedge \neg V(x)) \rightarrow \exists y(S(x,y) \wedge C(y)))$

$\{rm \rightarrow\}$ = $\forall x(\neg(E(x) \wedge \neg V(x)) \vee \exists y(S(x,y) \wedge C(y)))$

$\{prenex\}$ = $\forall x \exists y(\neg E(x) \vee V(x) \vee (S(x,y) \wedge C(y)))$

$\{skol\}$ = $\forall x(\neg E(x) \vee V(x) \vee (S(x,f(x)) \wedge C(f(x))))$

$\{add()\}$ = $\forall x((\neg E(x) \vee V(x)) \vee (S(x,f(x)) \wedge C(f(x))))$

$\{dist\}$ = $\forall x(\underbrace{(\neg E(x) \vee V(x) \vee S(x,f(x))}$
$\wedge \underbrace{(\neg E(x) \vee V(x) \vee C(f(x))))}$

Clauses:

(1a) $\neg E(x) \vee V(x) \vee S(x,f(x))$

(1b) $\neg E(x) \vee V(x) \vee C(f(x))$

$\forall x \exists y$ $\{y/f(x)\}$

Example: Standard Form (I)

(1)		$\forall x((E(x) \wedge \neg V(x)) \rightarrow \exists y(S(x,y) \wedge C(y)))$
$\{rm \rightarrow\}$	=	$\forall x(\neg(E(x) \wedge \neg V(x)) \vee \exists y(S(x,y) \wedge C(y)))$
$\{prenex\}$	=	$\forall x \exists y(\neg E(x) \vee V(x) \vee (S(x,y) \wedge C(y)))$
$\{skol\}$	=	$\forall x(\neg E(x) \vee V(x) \vee (S(x,f(x)) \wedge C(f(x))))$
CNF $\{add()\}$	=	$\forall x((\neg E(x) \vee V(x)) \vee (S(x,f(x)) \wedge C(f(x))))$
$\{dist\}$	=	$\forall x(\underbrace{(\neg E(x) \vee V(x) \vee S(x,f(x))}$
		$\wedge \underbrace{(\neg E(x) \vee V(x) \vee C(f(x)))})$

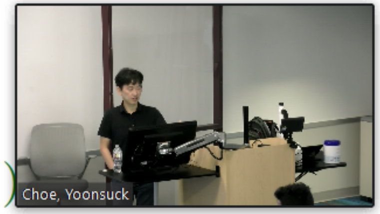
Clauses:

(1a) $\neg E(x) \vee V(x) \vee S(x,f(x))$

(1b) $\neg E(x) \vee V(x) \vee C(f(x))$

Eventually.

Example: Standard Form (I)



$$
\begin{array}{lcl}
(1) & & \forall x((E(x) \land \neg V(x)) \rightarrow \exists y(S(x,y) \land C(y))) \\
\{rm \ \rightarrow\} & = & \forall x(\neg(E(x) \land \neg V(x)) \lor \exists y(S(x,y) \land C(y))) \\
\{prenex\} & = & \forall x \exists y(\neg E(x) \lor V(x) \lor (S(x,y) \land C(y))) \\
\{skol\} & = & \forall x(\neg E(x) \lor V(x) \lor (S(x,f(x)) \land C(f(x)))) \\
\{add()\} & = & \forall x((\neg E(x) \lor V(x)) \lor (S(x,f(x)) \land C(f(x)))) \\
\{dist\} & = & \forall x(\underbrace{(\neg E(x) \lor V(x) \lor S(x,f(x))}) \\
 & & \quad \land \underbrace{(\neg E(x) \lor V(x) \lor C(f(x)))})
\end{array}
$$

assoc

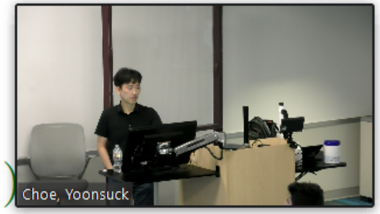
$\bigcirc \lor (\bigcirc \land \bigcirc)$

Clauses:

(1a) $\neg E(x) \lor V(x) \lor S(x,f(x))$

(1b) $\neg E(x) \lor V(x) \lor C(f(x))$

So this is in this phone. Then you can use the distributed law to get to these 2 closes



Example: Standard Form (I)

$$
\begin{aligned}
&(1) && \forall x((E(x) \wedge \neg V(x)) \rightarrow \exists y(S(x,y) \wedge C(y))) \\
&\{rm \rightarrow\} &=& \forall x(\neg(E(x) \wedge \neg V(x)) \vee \exists y(S(x,y) \wedge C(y))) \\
&\{prenex\} &=& \forall x \exists y(\neg E(x) \vee V(x) \vee (S(x,y) \wedge C(y))) \\
&\{skol\} &=& \forall x(\neg E(x) \vee V(x) \vee (S(x,f(x)) \wedge C(f(x)))) \\
&\{add()\} &=& \forall x((\neg E(x) \vee V(x)) \vee (S(x,f(x)) \wedge C(f(x)))) \\
&\{dist\} &=& \forall x(\underbrace{(\neg E(x) \vee V(x) \vee S(x,f(x))}) \\
& && \wedge \underbrace{(\neg E(x) \vee V(x) \vee C(f(x)))})
\end{aligned}
$$

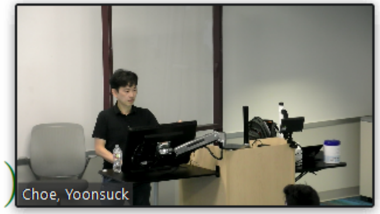
Two

Clauses:

(1a) $\neg E(x) \vee V(x) \vee S(x, f(x))$

(1b) $\neg E(x) \vee V(x) \vee C(f(x))$

Example: Standard Form (I)



$$
\begin{array}{lcl}
(1) & & \forall x((E(x) \wedge \neg V(x)) \rightarrow \exists y(S(x,y) \wedge C(y))) \\
\{rm \ \rightarrow\} & = & \forall x(\neg(E(x) \wedge \neg V(x)) \vee \exists y(S(x,y) \wedge C(y))) \\
\{prenex\} & = & \forall x \exists y(\neg E(x) \vee V(x) \vee (S(x,y) \wedge C(y))) \\
\{skol\} \ \checkmark & = & \forall x(\neg E(x) \vee V(x) \vee (S(x,f(x)) \wedge C(f(x)))) \\
\{add()\} & = & \forall x((\neg E(x) \vee V(x)) \vee (S(x,f(x)) \wedge C(f(x)))) \\
\{dist\} & = & \forall x(\underbrace{(\neg E(x) \vee V(x) \vee S(x,f(x))}) \\
 & & \quad \wedge \underbrace{(\neg E(x) \vee V(x) \vee C(f(x)))})
\end{array}
$$

Two

Clauses:

(1a) $\neg E(x) \vee V(x) \vee S(x, f(x))$

(1b) $\neg E(x) \vee V(x) \vee C(f(x))$

Now str those across 2 different closes

Example: Standard Form (III)

$$
\begin{array}{lcl}
(3) & & \forall x(D(x) \cancel{\rightarrow} \neg V(x)) \\
\{rm\ \rightarrow\} & = & \forall x(\neg D(x) \vee \neg V(x))
\end{array}
$$

Clause:

(3) $\neg D(x) \vee \neg V(x)$

$$
\begin{array}{lcl}
(4) & & \exists x(D(x) \wedge C(x)) \\
\{negate\} & \Rightarrow & \neg(\exists x(D(x) \wedge C(x))) \\
\{prenex\} & = & \forall x \neg(D(x) \wedge C(x))) \\
\{CNF\} & = & \forall x(\neg D(x) \vee \neg C(x))
\end{array}
$$

Clause:

(4) $\neg D(x) \vee \neg C(x)$

Example: English to First Order Logic

(1) The customs officials searched everyone who entered the country who was not a VIP. (2) Some of the drug dealers entered the country, and they were only searched by drug dealers. (3) No drug dealer was a VIP. (4) Some of the customs officials were drug dealers.

1. $\forall x((E(x) \land \neg V(x)) \rightarrow \exists y(S(x,y) \land C(y)))$
2. $\exists x(E(x) \land D(x) \land \forall y(S(x,y) \rightarrow D(y)))$
3. $\forall x(D(x) \rightarrow \neg V(x))$
4. $\exists x(D(x) \land C(x))$

Account
Dashboard
Courses
Calendar
Inbox
History
Commons
Help

CS.Fall.2022
Home
Syllabus
Pages
Files
Assignments
Grades
Accessibility Report
Core Curriculum Assessment
Course Evaluations
Mediasite Collection
Announcements
Modules
Discussions
People
Outcomes
Rubrics
Quizzes
Collaborations
Settings

CSCE 420-501: Artificial Intelligence

Syllabus: [Syllabus]

Quick links and info:

- Office hours:
  - Instructor: TF 2:30pm-4:00pm [Zoom]
  - TA: Jin Hyun Park, jinhyun.park@tamu.edu ; MWF 9:30am-11:00am, [Zoom]
  - Grader: Vicente Balmaseda, vibalcam@tamu.edu (grading related questions only)
- Lecture notes: [Lecture notes folder]
- Lecture recordings: [Link]
- Practice Exams: [TBA]; Exam Rules [Link]

NEWS:

- [09/26] Homework 2 announced [Link]
  - IMPORTANT: Homework 2 is due on Tuesday! Right before the midterm exam.
  - Homework 2 does not include programming.
- [09/26]: slide04-planning uploaded [Lecture notes folder]
- [09/04]: Practice exams uploaded [Link]
  - Note: solutions will not be provided.
- [08/31]: Homework 1 announced [Link]
- [08/30]: Philosopher AI sample results [Link]
- [08/25]:
  - Lecture notes and screenshots: [Lecture notes folder]
  - Lecture recordings: [Link]

Today

next Thu: midterm exam!

Fall Break

Weekly Schedule and Deadlines

Week	Tuesday	Thursday	Topic	Reading (3rd ed)	(4th ed)	Announcements	Dues	Lecture notes[Folder]	Comment
1	**no class**	8/25/2022	Introduction	Ch 1, 26.1, 26.2	Ch 1, 27.1, 27.2			slide01.pdf	

Import from Commons
Choose Home Page
View Course Stream
Course Setup Checklist
New Announcement
New Analytics
View Course Notifications

To Do

92 Grade Homework 1
100 points • Sep 25 at 11:59pm

2 Grade Homework 2
100 points • Oct 11 at 11:59pm

HW#1 Solution TBA: this week.

Coming Up View Calendar

Nothing for the next week

You can go there, hold on, name, did I say?

You are screen sharing | Stop Share

Type here to search
11:11 AM
10/4/2022

Example: Standard Form (III)

(3)		$\forall x(D(x) \rightarrow \neg V(x))$
$\{rm\ \rightarrow\}$	=	$\forall x(\neg D(x) \vee \neg V(x))$

Clause:

(3) $\neg D(x) \vee \neg V(x)$

(4)		$\exists x(D(x) \wedge C(x))$
$\{negate\}$	$\Rightarrow$	$\neg(\exists x(D(x) \wedge C(x)))$
$\{prenex\}$	=	$\forall x \neg(D(x) \wedge C(x)))$
$\{CNF\}$	=	$\forall x(\neg D(x) \vee \neg C(x))$

Clause:

(4) $\neg D(x) \vee \neg C(x)$

Of X or not, c. of X. So in this case you just have a single clause as a result of negation of the final wow.

Example: Clauses

(1a) $\neg E(x) \lor V(x) \lor S(x, f(x))$

(1b) $\neg E(x) \lor V(x) \lor C(f(x))$

(2a) $E(a)$

(2b) $D(a)$

(2c) $\neg S(a, y) \lor D(y)$

(3) $\neg D(x) \lor \neg V(x)$

(4) $\neg D(x) \lor \neg C(x)$

$\sigma = \{x/a\}$

(5) $V(a) \lor S(a, f(a))$

(6) $V(a) \lor C(f(a))$

(7) $D(f(a))$

(8) $V(a) \lor \neg D(f(a))$

$\sigma = \{y/f(a)\}$

$\sigma = \{x/f(a)\}$

Note: The input to your theorem prover will be in a standard form like the above.

Exercise 1: rewrite the above in LISP representation.

Exercise 2: use resolution to derive $\mathbf{False}$.

The of fBI, because you replace X with a V. Then next

Example: Clauses

(1a) $\neg E(x) \vee V(x) \vee S(x, f(x))$

(1b) $\neg E(x) \vee V(x) \vee C(f(x))$

(2a) $E(a)$

(2b) $D(a)$

(2c) $\neg S(a, y) \vee D(y)$

(3) $\neg D(x) \vee \neg V(x)$

(4) $\neg D(x) \vee \neg C(x)$

Note: The input to your theorem prover will be in a standard form like the above.

Exercise 1: rewrite the above in LISP representation.

Exercise 2: use resolution to derive **False** .

$\sigma = \{x/a\}$

(5) $V(a) \vee S(a, f(a))$

(6) $V(a) \vee C(f(a))$

(7) $D(f(a))$

$\sigma = \{y/f(a)\}$

(8) $V(a) \vee \neg D(f(a))$

$\sigma = \{x/f(a)\}$

(9) $V(a)$

(10) $\neg D(a)$

$\sigma = \{x/a\}$

(11) False

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

Choe, Yoonsuck

① C_4: $S(a,y) \vee D(y)$ $\sigma = \{x/y\}$

C_3: $\neg D(x) \vee \neg V(x)$

② $\neg S(a,y) \vee \neg V(y)$

Example: Clauses

$\sigma = \{x/a\}$

(1a) $\neg E(x) \vee V(x) \vee S(x, f(x))$

(1b) $\neg E(x) \vee V(x) \vee C(f(x))$

(2a) $E(a)$

(2b) $D(a)$

(2c) $\neg S(a,y) \vee D(y)$

(3) $\neg D(x) \vee \neg V(x)$

(4) $\neg D(x) \vee \neg C(x)$

(5) $V(a) \vee S(a, f(a))$

(6) $V(a) \vee C(f(a))$

(7) $D(f(a))$ $\sigma = \{y/f(a)\}$

(8) $V(a) \vee \neg D(f(a))$ $\sigma = \{x/f(a)\}$

Note: The input to your theorem prover will be in a standard form like the above.

Exercise 1: rewrite the above in LISP representation.

(9) $V(a)$

Exercise 2: use resolution to derive **False** .

(10) $\neg D(a)$ $\sigma = \{x/a\}$

$\sigma = \{y/a\}$

$S(x, f(x))$

(11) False

$\neg S(a,a)$

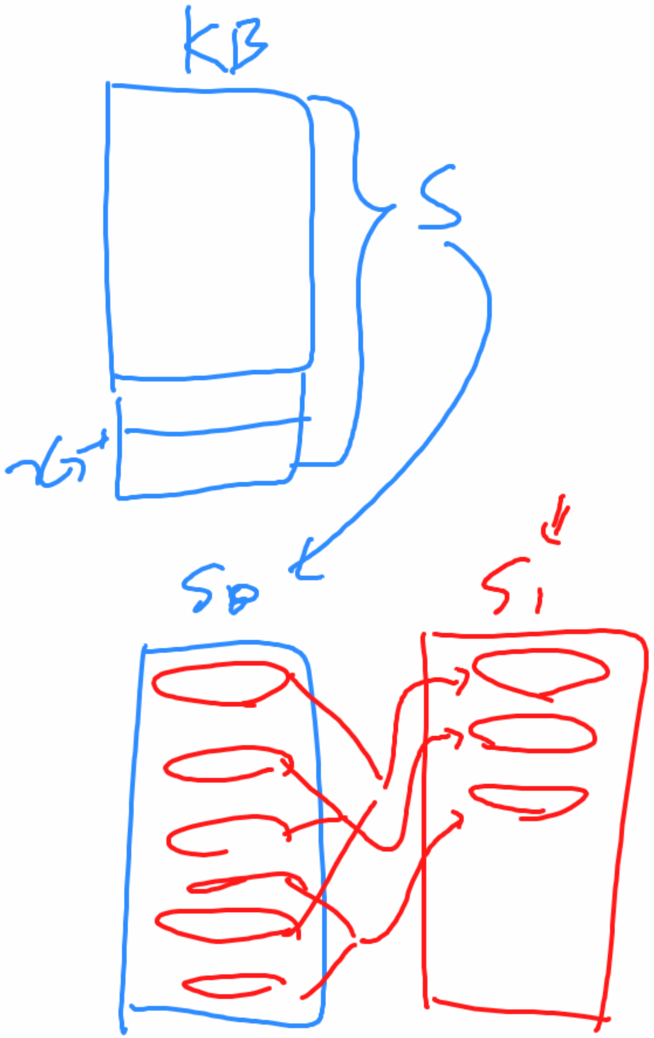
123

Yeah. But if you implement this in in a program and assist to do this all in a very systematic manner, then expressions like this would be easily generated.

You are screen sharing   Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On



Basic Theorem Proving Algorithm

Level saturation resolution method (or two-pointer method)

Generate all possible resolvents:

- Generate sequences of clause sets $S^0, S^1, S^2, \ldots$, where

  $S^0 \quad = \quad S$ (original set of clauses)

  $S^n \quad = \quad \{$ all possible resolvents of clauses

  $C_1 \in (S^0 \cup \ldots S^{n-1})$ and $C_2 \in S^{n-1}$ $\}$

- This is basically a **breadth first search** method, and it can be extremely inefficient except for small problems.
- The problem is that **irrelevant** derivations are made: in generating an n-step proof, we also generate **all possible derivations** of n-1 steps.

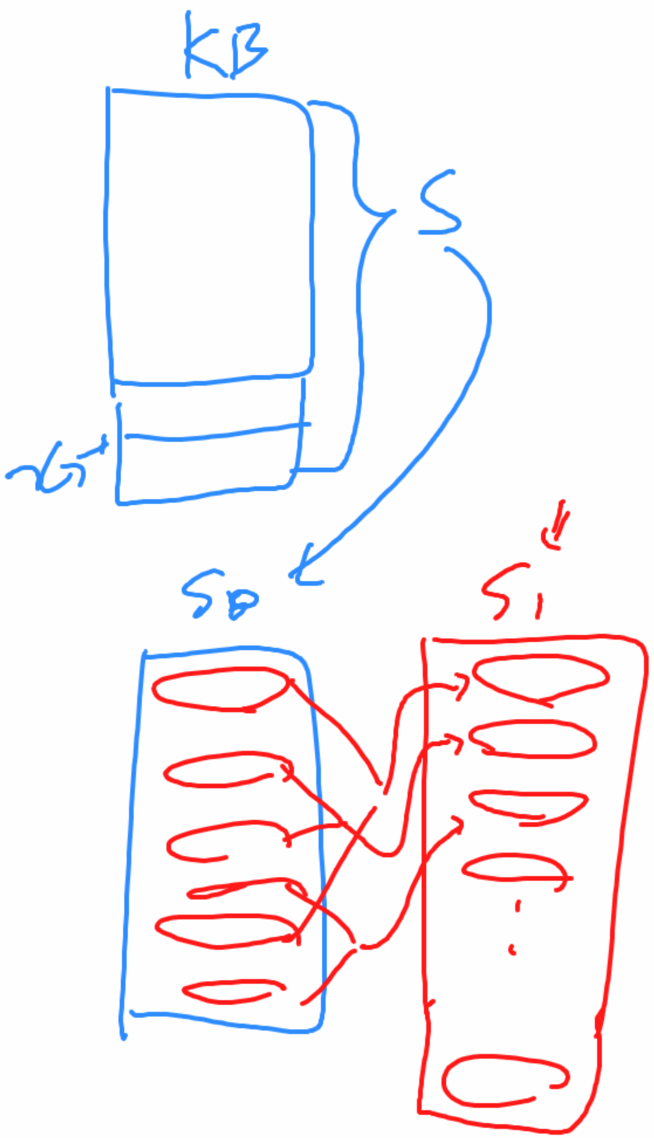
$C_2 \in S^{1-1} = S^0$

$n=1$ $\quad$ $n=1$

$C_1 \in S^0 \cup \cdots \cup S^{1-1} = S^0 \cup \cdots \cup S^0 = S^0$

0 resolve it. let's say these 2 are resolvable, and you plug it in

You are screen sharing Stop Share



Basic Theorem Proving Algorithm

Level saturation resolution method (or two-pointer method)

Generate all possible resolvents:

- Generate sequences of clause sets $S^0, S^1, S^2, \ldots$, where

  $S^0 \quad = \quad S$ (original set of clauses)

  $S^n \quad = \quad \{$ all possible resolvents of clauses $C_1 \in (S^0 \cup \ldots S^{n-1})$ and $C_2 \in S^{n-1}$ $\}$

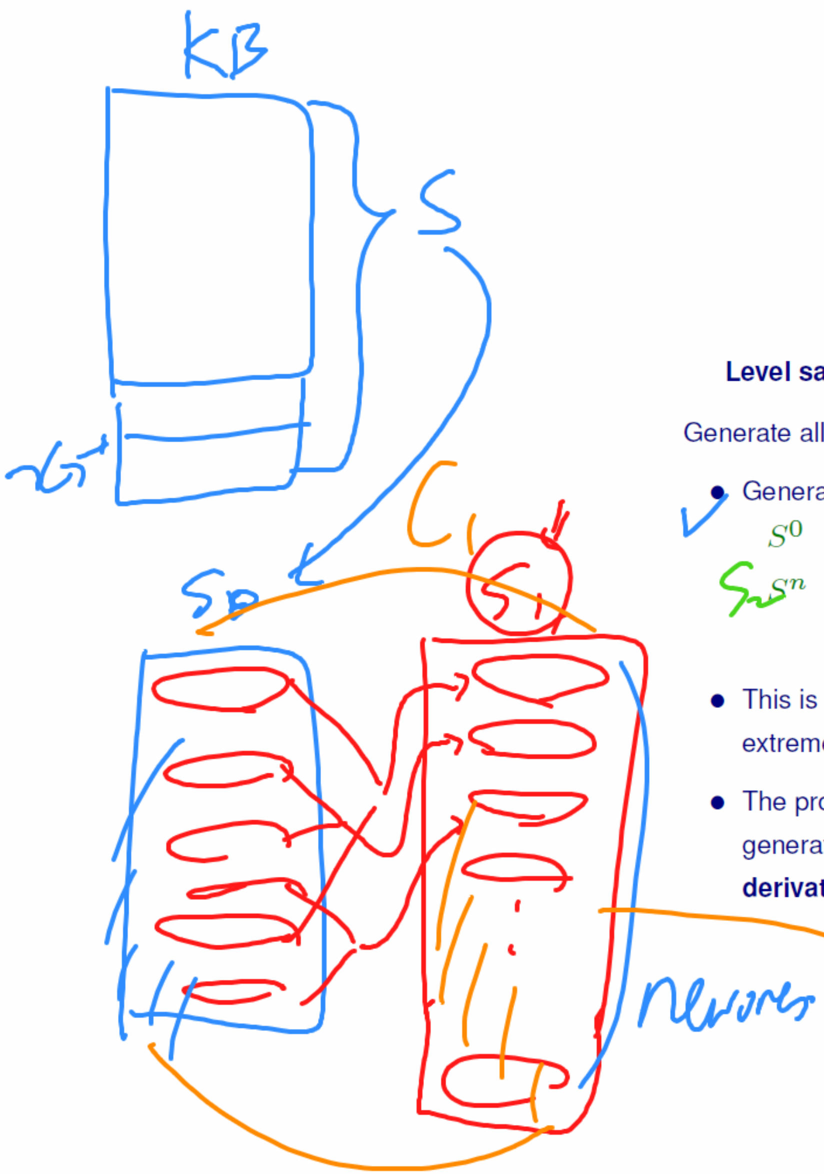
- This is basically a **breadth first search** method, and it can be extremely inefficient except for small problems.
- The problem is that **irrelevant** derivations are made: in generating an n-step proof, we also generate **all possible derivations** of n-1 steps.

$C_2 \in S^{1-1} = S^0$

$n=1$ $n=1$

$C_1 \in S^0 \cup \cdots \cup S^{1-1} = S^0 \cup \cdots \cup S^0 = S^0$

Then s, 2,



Basic Theorem Proving Algorithm

Level saturation resolution method (or two-pointer method)

Generate all possible resolvents:

- Generate sequences of clause sets $S^0, S^1, S^2, \ldots$, where

$$S^0 = S \quad \text{(original set of clauses)}$$

S_n $$S^n = \{ \text{ all possible resolvents of clauses } C_1 \in (S^0 \cup \ldots S^{n-1}) \text{ and } C_2 \in S^{n-1} \}$$

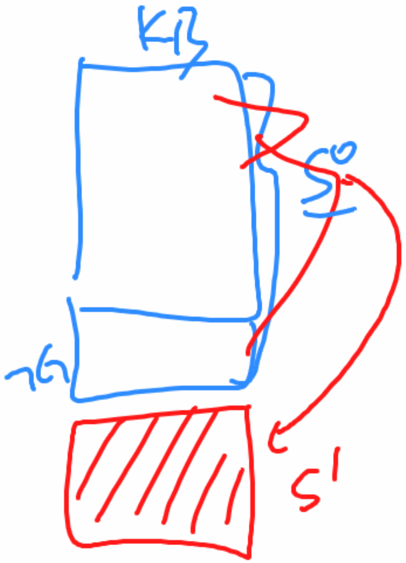
- This is basically a **breadth first search** method, and it can be extremely inefficient except for small problems.
- The problem is that **irrelevant** derivations are made: in generating an n-step proof, we also generate **all possible derivations** of n-1 steps.

S_2 :

$C_1 \in S^0 \cup \ldots \cup S^{2-1}$

$= S^0 \cup S^1$

$C_2 \in S^{2-1} = S^1$



Basic Theorem Proving Algorithm

Level saturation resolution method (or two-pointer method)

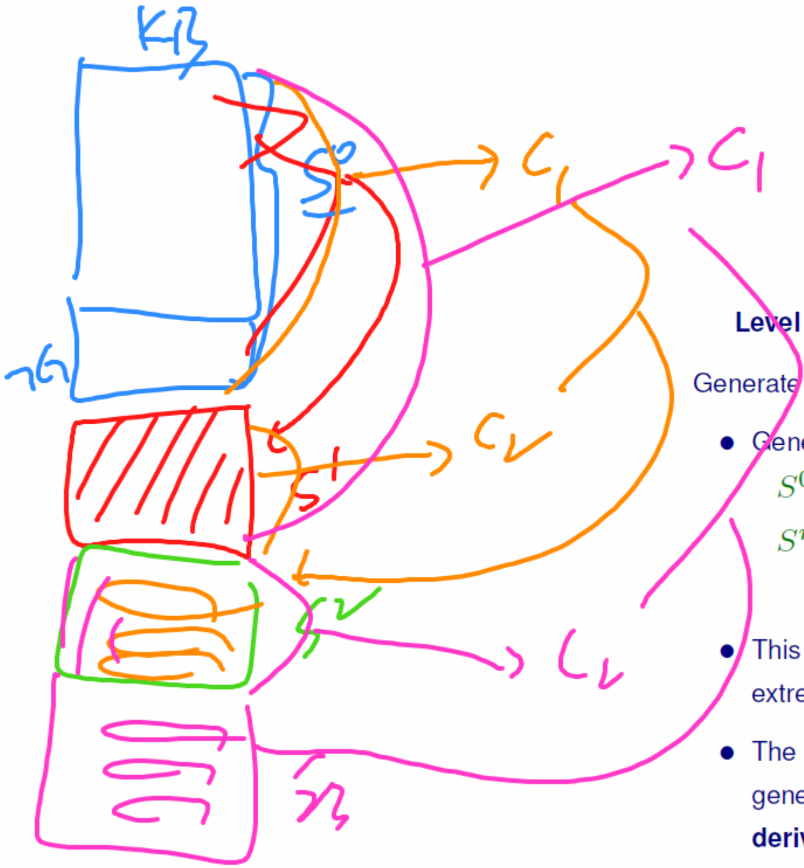
Generate all possible resolvents:

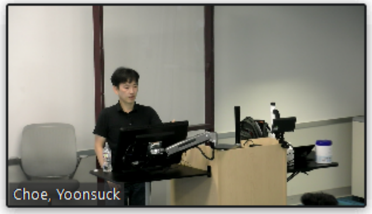
- Generate sequences of clause sets $S^0, S^1, S^2, \ldots$, where

$$\begin{aligned} S^0 &= S \quad \text{(original set of clauses)} \\ S^n &= \{ \text{ all possible resolvents of clauses} \\ &\quad\;\; C_1 \in (S^0 \cup \ldots S^{n-1}) \text{ and } C_2 \in S^{n-1} \; \} \end{aligned}$$

- This is basically a **breadth first search** method, and it can be extremely inefficient except for small problems.
- The problem is that **irrelevant** derivations are made: in generating an n-step proof, we also generate **all possible derivations** of n-1 steps.

So this would be now s one, and this is assume





Basic Theorem Proving Algorithm

Level saturation resolution method (or two-pointer method)

Generate all possible resolvents:

- Generate sequences of clause sets $S^0, S^1, S^2, \ldots$, where

$$S^0 \quad = \quad S \quad \text{(original set of clauses)}$$

$$S^n \quad = \quad \{ \text{ all possible resolvents of clauses } C_1 \in (S^0 \cup \ldots S^{n-1}) \text{ and } C_2 \in S^{n-1} \ \}$$

- This is basically a **breadth first search** method, and it can be extremely inefficient except for small problems.
- The problem is that **irrelevant** derivations are made: in generating an n-step proof, we also generate **all possible derivations** of n-1 steps.

From the immediate resulting one then there are these so that's the that's the algorithm

Deletion Strategy

To reduce the huge number of generated clauses, we would like to delete clauses whenever possible. We can delete:

1. Any tautology, e.g. $P(a) \vee \neg P(a) \vee Q(x)$.
2. Any clause which duplicates an existing clause.
3. Any clause which is **subsumed by** an existing clause.

A clause C **subsumes** a clause D iff there is a substitution σ such that $C\sigma \subseteq D$ (recall that a clause can be represented as a set of literals). D is called a **subsumed clause**.

Deletion strategy will be complete if it is used with certain resolution algorithms (such as level saturation).

22 FALL CSCE 420 501: ARTIFICIA | Launch Meeting - Zoom

canvas.tamu.edu/courses/171555

22 FALL CSCE 420 5... | 22 FALL CSCE 633 6... | Video Conferencing...

Mouse | Select | Text | Draw | Stamp | Spotlight | Eraser | Format | Undo | Redo | Clear | Save

Who can see what you share here? Recording On

Account | Dashboard | Courses | Calendar | Inbox | History | Commons | Help

CS.Fall.2022

- Home
- Syllabus
- Pages
- Files
- Assignments
- Grades
- Accessibility Report
- Core Curriculum Assessment
- Course Evaluations
- Mediasite Collection
- Announcements
- Modules
- Discussions
- People
- Outcomes
- Rubrics
- Quizzes
- Collaborations
- Settings

			(informed)	4.3	4.3			gameplaying.pdf	
4	9/13/2022	9/15/2022	Game playing	Ch 5	Ch 5			slide02-search-and-gameplaying.pdf	
5	9/20/2022	9/22/2022	Propositional logic and theorem proving	Ch 7	Ch 7	hw2 announced	hw1 due 9/25 11:59pm	slide03-logic	
6	9/27/2022	9/29/2022	First-order logic and theorem proving	Ch 8, 9	Ch 8, 9			slide03-logic	
7	10/4/2022	10/6/2022	First-order logic and theorem proving; Prolog; Planning;	Ch 8, 9	Ch 8, 9			slide03-logic, slide04-planning	
8	no class	10/13/2022 **Midterm Exam**	**Midterm Exam (Thu, in class)**	Ch 10	Ch 11	hw3 announced	hw2 due: 10/11 Tuesday 11:59pm!!		10/10: midsemester grades due
9	10/18/2022	10/20/2022	Uncertainty and probabilistic reasoning	Ch 13, 14	Ch 12, 13				
10	10/25/2022	10/27/2022	Uncertainty and probabilistic reasoning	Ch 13, 14	Ch 12, 13				
11	11/1/2022	11/3/2022	Machine learning intro	Ch 18.1-18.2	Ch 19.1-19.2	hw4 announced	hw3 due		
12	11/8/2022	11/10/2022	Machine learning : decision tree	Ch 18.3	Ch 19.3				
			Machine learning:		Ch				11/18: Q-

Today

Fall Break

next Thu: midterm exam!

HW#1 Solution TBA: this week.

Do this.

You are screen sharing | Stop Share

Type here to search

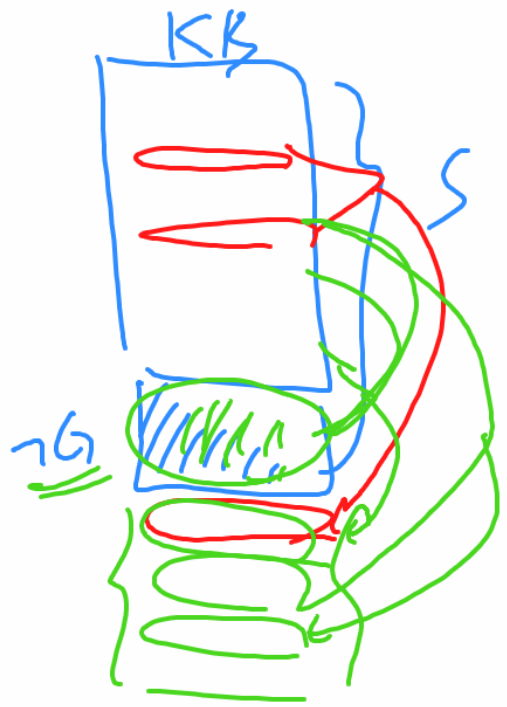
11:11 AM
10/4/2022

Subsumed Clause: Example (I)

Skip

Example:

- $C = P(x)$ $\sigma = \{x/a\}$
- $D = P(a) \vee Q(a)$
- If $\sigma = \{x/a\}$, then
  $C\sigma = P(a) = \{P(a)\}$
  $\subseteq \{P(a), Q(a)\} = P(a) \vee Q(a) = D$.
- Since $C\sigma \subseteq Q$, C subsumes D, and D can be deleted.



Strategies to Improve Resolution

1. **Deletion strategy**: remove tautology, duplicates, and subsumed clauses.
2. **Unit preference**: resolve with clauses with the fewest literals.
3. **Set of support**: begin with set T consisting of the clauses from the **negated conclusion**. Each resolution step must involve a member of T, and the result is added to T.
4. **Linear resolution** (Depth First): Each step must be a resolution step involving the clause produced by the last step.

But again, now you have all of the issues with steps for search so it's not complete it's not up

Advantages and Disadvantages of Resolution

- **Advantages:** (1) Resolution is universally applicable to problems which can be described in first-order logic. (2) The theorem proving engine can be decoupled from any particular domain.

Strength & Weakness

- **Disadvantage:** (1) Resolution is too inefficient to be generally applicable. (2) This is partly because resolution is purely syntactic, and it does not consider what the predicates *mean*. For this reason, developing a domain-dependent heuristic is impossible. (3) A contradiction in the axiom set may allow anything to be proved. (4) It is difficult for a human to understand proof by resolution prover.

And finally, it's very difficult to understand the proof by resolution, approval for humans to see a bunch of these resolutions going on

Question Answering: Example

Example:

KB

Rules

1. $\forall x \forall y \forall z((Parent(x, z) \wedge Parent(z, y)) \rightarrow Grandparent(x, y)$
2. $\forall x \forall y(Mother(x, y) \rightarrow Parent(x, y))$
3. $\forall x \forall y(Father(x, y) \rightarrow Parent(x, y))$

Facts

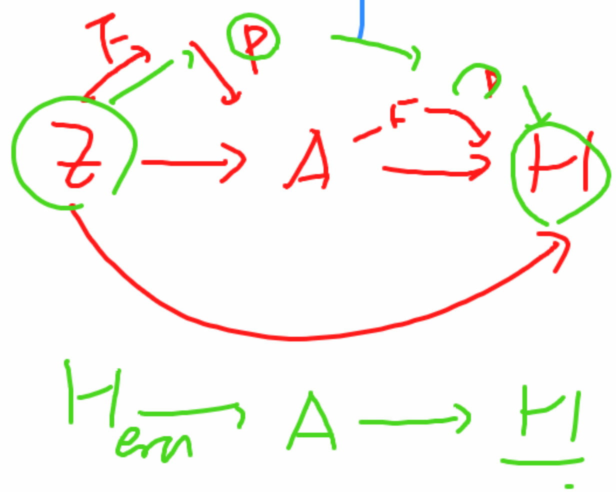
4. $Father(Zeus, Ares)$
5. $Mother(Hera, Ares)$
6. $Father(Ares, Harmonia)$

Question: "Who is a grandparent of Harmonia?"

1. $\exists x(Grandparent(x, Harmonia))$

   Negated: $\neg \exists x(Grandparent(x, Harmonia))$

   $= \forall x(\neg Grandparent(x, Harmonia))$



Question Answering: Example

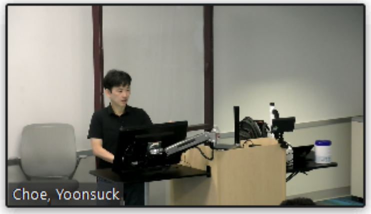
Example:

1. $\forall x \forall y \forall z((Parent(x,z) \land Parent(z,y)) \rightarrow Grandparent(x,y)$
2. $\forall x \forall y(Mother(x,y) \rightarrow Parent(x,y))$
3. $\forall x \forall y(Father(x,y) \rightarrow Parent(x,y))$
4. $Father(Zeus, Ares)$
5. $Mother(Hera, Ares)$
6. $Father(Ares, Harmonia)$

Skolemization

Question: "Who is a grandparent of Harmonia?"

1. $\exists x(Grandparent(x, Harmonia))$
   Negated: $\neg \exists x(Grandparent(x, Harmonia))$
   $= \forall x(\neg Grandparent(x, Harmonia))$



Question Answering: Result

- Resolution on the previous example generates $\mathbf{False}$ in the end, but what that answers is the question **“Is there a grandparent of Harmonia?”**. Of course the answer is **yes**, but the question is **who?**.
- The negated question in the above examples was $\neg Grandparent(x, Harmonia)$. Clearly, the binding which x ultimately receives is the desired answer!
- Observation: one substitution along the way, starting from $\neg Grandparent(x, Harmonia)$, the negated conclusion, is $\{x/Hera\}$, thus $Hera$ must be an answer.

Exercise: use resolution to derive $\mathbf{False}$ in the example in the previous slide.

Answer Extraction

We can introduce special predicates to extract the answers.

- **Answer predicate:**

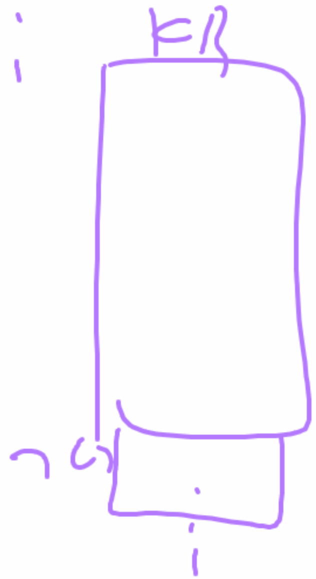
  Hera

  $\neg Grandparent(x, Harmonia) \vee Answer(x)$ x, Hera

- The answer predicate has these properties:
  - It does not resolve with anything, but it keeps track of variable bindings.
  - The theorem prover recognize a clause consisting only of the $Answer$ predicate as **False** .
- For example, resolution on the previous example results in:

$$Answer(Hera)$$

  as the final clause.



$\boxed{Answer}(Hera)$

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

① Add facts first

↳ group then. F→M→F ✗

F→F→M ○

Question Answering: Example

Example:

1. $\forall x \forall y \forall z ((Parent(x,z) \wedge Parent(z,y)) \rightarrow Grandparent(x,y))$
2. $\forall x \forall y (Mother(x,y) \rightarrow Parent(x,y$
3. $\forall x \forall y (Father(x,y) \rightarrow Parent(x,y))$
4. $Father(Zeus, Ares)$
5. $Mother(Hera, Ares)$
6. $Father(Ares, Harmonia)$

var: X

const, pred, fn: x

def predicates used later first.

Question: "Who is a grandparent of Harmonia?"

1. $\exists x (Grandparent(x, Harmonia))$

Negated: $\neg \exists x (Grandparent(x, Harmo$

$= \forall x (\neg Grandparent(x, Harmonia$

f. M

↳ p

linux2.cse.tamu.edu - PuTTY

```
> refer to Texas A&M University System Policy 29.01.03.
|
| To report an issue with this machine please email: linux-engr-hel
> du and include your department, netid, computer location and comp
| ****************************************************************
> ***********************
|
| If you get the error message: Failed to create home directory
| Then you will need to first create your home directory. Directio
> ble in your browser by visiting the following web page: http://bi
> medoc
|
| COMPUTER NAME: linux2.cs.tamu.edu
|
 End of banner message from server
 choe@sun.cs.tamu.edu's password:
linux2:/home/faculty/c/choe> gprolog
GNU Prolog 1.4.4 (64 bits)
Compiled Jan 15 2014, 20:38:19 with gcc
By Daniel Diaz
Copyright (C) 1999-2013 Daniel Diaz
| ?- [user].
compiling user for byte code...
```

First, Hey, father, mother, then define parents. So these are some basic rules

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

① Add facts

$A \vee B \rightarrow C$

Question Answering: Example

$C \quad \neg \quad A \quad \vee \quad B.$

it on

Example:

1. $\forall x \forall y \forall z ((Parent(x, z) \wedge Parent(z, y)) \rightarrow Grandparent(x, y)$
2. $\forall x \forall y (Mother(x, y) \rightarrow Parent(x, y$
3. $\forall x \forall y (Father(x, y) \rightarrow Parent(x, y)$
4. $Father(Zeus, Ares)$
5. $Mother(Hera, Ares)$
6. $Father(Ares, Harmonia)$

Question: "Who is a grandparent of Harmonia?"

1. $\exists x (Grandparent(x, Harmonia))$

   Negated: $\neg \exists x (Grandparent(x, Harmo$

   $= \forall x (\neg Grandparent(x, Harmonia$

linux2.cse.tamu.edu - PuTTY

```
> refer to Texas A&M University System Policy 29.01.03.
|
| To report an issue with this machine please email: linux-engr-hel
> du and include your department, netid, computer location and comp
| ****************************************************************
> ***********************
|
| If you get the error message: Failed to create home directory
| Then you will need to first create your home directory. Direction
> ble in your browser by visiting the following web page: http://bi
> medoc
|
| COMPUTER NAME: linux2.cs.tamu.edu
|
  End of banner message from server
  choe@sun.cs.tamu.edu's password:
linux2:/home/faculty/c/choe> gprolog
GNU Prolog 1.4.4 (64 bits)
Compiled Jan 15 2014, 20:38:19 with gcc
By Daniel Diaz
Copyright (C) 1999-2013 Daniel Diaz
| ?- [user].
compiling user for byte code...
```


And then this is all

You are screen sharing Stop Share

22 FALL CSCE 420 501: ARTIFICIA × Launch Meeting - Zoom × +

canvas.tamu.edu/courses/171555

22 FALL CSCE 420 5... 22 FALL CSCE 633 6... Video Conferencing...

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

Account
Dashboard
Courses
Calendar
Inbox 7
History
Commons
Help 10

CS.Fall.2022

- Home
- Syllabus
- Pages
- Files
- Assignments
- Grades
- Accessibility Report
- Core Curriculum Assessment
- Course Evaluations
- Mediasite Collection
- Announcements
- Modules
- Discussions
- People
- Outcomes
- Rubrics
- Quizzes
- Collaborations
- Settings

Quick links and info:

- Office hours:
  - Instructor: TF 2:30pm-4:00pm [Zoom]
  - TA: Jin Hyun Park, jinhyun.park@tamu.edu ; MWF 9:30am-11:00am, [Zoom]
  - Grader: Vicente Balmaseda, vibalcam@tamu.edu (grading related questions only)
- Lecture notes: [Lecture notes folder]
- Lecture recordings: [Link]
- Practice Exams: [TBA]; Exam Rules [Link]

NEWS:

- [09/26] Homework 2 announced [Link]
  - IMPORTANT: Homework 2 is due on Tuesday! Right before the midterm exam.
  - Homework 2 does not include programming.
- [09/26]: slide04-planning uploaded [Lecture notes folder]
- [09/04]: Practice exams uploaded [Link]
  - Note: solutions will not be provided.
- [08/31]: Homework 1 announced [Link]
- [08/30]: Philosopher AI sample results [Link]
- [08/25]:
  - Lecture notes and screenshots: [Lecture notes folder]
  - Lecture recordings: [Link]

Weekly Schedule and Deadlines

Week	Tuesday	Thursday	Topic	Reading (3rd ed)	(4th ed)	Announcements	Dues	Lecture notes[Folder]	Comment
1	**no class**	8/25/2022	Introduction	Ch 1, 26.1, 26.2	Ch 1, 27.1, 27.2			slide01.pdf	
2	8/30/2022	9/1/2022	Search (uninformed)	Ch 3.1-3.6	Ch 3.1-3.6	hw1 announced		slide02-search-and-gameplaying.pdf	8/30: last day for add/drop

Okay.

View Course Stream
Course Setup Checklist
New Announcement
New Analytics
View Course Notifications

To Do

92 Grade Homework 1
100 points • Sep 25 at 11:59pm

2 Grade Homework 2
100 points • Oct 11 at 11:59pm

Coming Up View Calendar

Nothing for the next week

You are screen sharing Stop Share

https://canvas.tamu.edu/courses/171555/gradebook/speed_grader?assignment_id=1425199

Type here to search

11:12 AM
10/4/2022

22 FALL CSCE 420 501: ARTIFICIA | Launch Meeting - Zoom

canvas.tamu.edu/courses/171555

22 FALL CSCE 420 5... | 22 FALL CSCE 633 6... | Video Conferencing...

Mouse | Select | Text | Draw | Stamp | Spotlight | Eraser | Format | Undo | Redo | Clear | Save

Who can see what you share here? Recording On

Account | Dashboard | Courses | Calendar | Inbox 7 | History | Commons | Help 10

CS.Fall.2022

- Home
- Syllabus
- Pages
- Files
- Assignments
- Grades
- Accessibility Report
- Core Curriculum Assessment
- Course Evaluations
- Mediasite Collection
- Announcements
- Modules
- Discussions
- People
- Outcomes
- Rubrics
- Quizzes
- Collaborations
- Settings

Quick links and info:

- Office hours:
  - Instructor: TF 2:30pm-4:00pm [Zoom]
  - TA: Jin Hyun Park, jinhyun.park@tamu.edu ; MWF 9:30am-11:00am, [Zoom]
  - Grader: Vicente Balmaseda, vibalcam@tamu.edu (grading related questions only)
- Lecture notes: [Lecture notes folder]
- Lecture recordings: [Link]
- Practice Exams: [TBA]; Exam Rules [Link]

NEWS:

- [09/26] Homework 2 announced [Link]
  - IMPORTANT: Homework 2 is due on Tuesday! Right before the midterm exam.
  - Homework 2 does not include programming.
- [09/26]: slide04-planning uploaded [Lecture notes folder]
- [09/04]: Practice exams uploaded [Link]
  - Note: solutions will not be provided.
- [08/31]: Homework 1 announced [Link]
- [08/30]: Philosopher AI sample results [Link]
- [08/25]:
  - Lecture notes and screenshots: [Lecture notes folder]
  - Lecture recordings: [Link]

Weekly Schedule and Deadlines

Week	Tuesday	Thursday	Topic	Reading (3rd ed)	(4th ed)	Announcements	Dues	Lecture notes[Folder]	Comment
1	**no class**	8/25/2022	Introduction	Ch 1, 26.1, 26.2	Ch 1, 27.1, 27.2			slide01.pdf	
2	8/30/2022	9/1/2022	Search (uninformed)	Ch 3.1-3.6	Ch 3.1-3.6	hw		slide02-search-gameplaying.pdf	8/30: last add/drop

View Course Stream
Course Setup Checklist
New Announcement
New Analytics
View Course Notifications

To Do

- Grade Homework 1 (92) — 100 points • Sep 25 at 11:59pm
- Grade Homework 2 (2) — 100 points • Oct 11 at 11:59pm

Coming Up — View Calendar

Nothing for the next week

And also it's very important to read the exam rules very very carefully, and

https://canvas.tamu.edu/courses/171555/gradebook/speed_grader?assignment_id=1425199

You are screen sharing | Stop Share

Type here to search | 11:12 AM 10/4/2022

CS.Fall.2022
Home
Syllabus
Pages
Files
Assignments
Grades
Accessibility Report
Core Curriculum Assessment
Course Evaluations
Mediasite Collection
Announcements
Modules
Discussions
People
Outcomes
Rubrics
Quizzes
Collaborations
Settings

Exam Rules

Please read these instructions very carefully.

1. Bring your TAMU ID or driver's license. You will not be allowed to take the exam without your ID.
2. This is a closed book exam. Phone / calculator use is prohibited. Put those in your bag.
3. You may prepare 1 sheet (US letter size = printer paper, both sides) of hand-written notes, for reference during the exam
   - PLEASE use US letter paper (printer paper: 11in x 8.5in). Do not use pages ripped off from your notebook.
   - Must be hand-written! Printouts, photocopies will be confiscated.
     - Exception: notes directly hand-written on iPad or tablet with reasonable text size may be printed out and used.
   - Submit your notes with your exam.
   - Write your name at the top left corner of both sides.
     - Notes without the name will be confiscated.
4. Blank scratch paper will be provided. You are only allowed to use the provided scratch paper.
   - DO NOT write your answers on the scratch paper. Those will not be graded.
5. Exam materials :
   - Exam materials will only be from the lecture notes and notes written on the screen during the lecture (screenshots), and from the homeworks.
   - Midterm: materials up to and including those covered by the lecture prior to the midterm exam.
   - Final exam: It will not be cumulative. Materials will be those covered after the midterm exam.
6. Practice exams:
   - Some materials on the practice exams have not been covered during this semester. These will not be on the exam.

Details are here, and extend material will be as I

Week	Tuesday	Thursday	Topic	Reading (3rd ed)	(4th ed)	Announcements	Dues	Lecture notes[Folder]	Comment
1	**no class**	8/25/2022	Introduction	Ch 1, 26.1, 26.2	Ch 1, 27.1, 27.2			slide01.pdf	
2	8/30/2022	9/1/2022	Search (uninformed)	Ch 3.1-3.6	Ch 3.1-3.6	hw1 announced		slide02-search-and-gameplaying.pdf	8/30: last day for add/drop
3	9/6/2022	9/8/2022	Search (informed)	Ch 4.1-4.3	Ch 4.1-4.3			slide02-search-and-gameplaying.pdf	
4	9/13/2022	9/15/2022	Game playing	Ch 5	Ch 5			slide02-search-and-gameplaying.pdf	
5	9/20/2022	9/22/2022	Propositional logic and theorem proving	Ch 7	Ch 7	hw2 announced	hw1 due 9/25 11:59pm	slide03-logic	
6	9/27/2022	9/29/2022	First-order logic and theorem proving	Ch 8, 9	Ch 8, 9			slide03-logic	
7	10/4/2022	10/6/2022	First-order logic and theorem proving; Prolog ; Planning;	Ch 8, 9	Ch 8, 9			slide03-logic, slide04-planning	
8	**no class**	10/13/2022 **Midterm Exam**	**Midterm Exam (Thu, in class)**	Ch 10	Ch 11	hw3 announced	hw2 due: 10/11 Tuesday 11:59pm!!		10/10: midsemester grades due
9	10/18/2022	10/20/2022	Uncertainty and probabilistic	Ch 13, 14	Ch 12, 13				

exam range
slide01
~ slide03 (propositional logic)

I'll send out a separate announcement just to put it on the back