Set-based Search Example: Resolution

CPSC 433: Artificial Intelligence
Fall 2024

Jonathan Hudson, Ph.D.
Assistant Professor (Teaching)
Department of Computer Science
University of Calgary

August 8, 2024

UNIVERSITY OF CALGARY

Logic Review

Review of Logic Definitions

- **Propositional logic (*zeroth order logic*)**
  - does not have predicates, just formulas of singular propositional symbols,
  - often p,q,r,… combined with (or ∨, and ∧, not ¬, implication →, biconditional ↔)
  - Ex. $\neg p \lor q \to r$

- **First-order logic**
  - formulas use variables, constants, predicates, functions
  - quantifier ∃ (there is)
  - quantifier ∀ (for all)
  - equality also possible (EQ)

Review of Logic Definitions

- **Variable**
  - generally w,x,y,z

- **Constant**
  - generally a,b,c,d,….
  - Or sometimes alice, bob, carol, etc. or similar.
  - Can replace a variable

Review of Logic Definitions

- **Predicate**
  - a property or relation
  - generally P, Q, R, etc.
  - P(a) would mean a constant a has property P
  - while P(x) would mean the same for indeterminate variable
  - **returns truth value**

- **Function**
  - constants are a subset of these with no parameters
  - generally f, g, h, etc.
  - maps within domain of variables
  - f(x) -> y where both x, y are in domain of problem

Review of Logic Definitions

- Clause
  - a single logical formula
- Disjunction
  - or
  - ∨
- Conjunction
  - And
  - ∧
- Negation
  - Not
  - ¬

Review of Logic Definitions

- Conjunctive Normal Form (CNF)
  - a set of clauses changed to a form where it becomes a conjunction of clauses where each clause is a disjunction of literals
  - Have clauses A, B, C
    - then conjunction of them becomes A and B and C
  - Every formula can be written in this form.
  - Note negations and brackets are transformed by logical rules such that negations apply to predicates and brackets are around clauses
  - $\neg(B \vee C)$ becomes $(\neg B) \wedge (\neg C)$
  - $(\mathrm{A} \wedge B) \vee C$ becomes $(A \vee C) \wedge (B \vee C)$

UNIVERSITY OF CALGARY

Review of Logic Definitions

- Unification
  - in our case used to attempt to find the most general unifier (**mgu**)
  - **mgu** is a valid mapping of variable/constant/function mapping to make two terms the same
  - Ex. if I have f(a) and f(x)
    - mgu mapping **x->a** makes f(a)=f(a)
- Resolution
  - theorem proving technique, general process is to

1. Take known clauses and **negate the conclusion trying to be proven(!)**
2. Then turn this into CNF
3. Attempt to derive empty clause
4. If found this indicates the set of clauses was not satisfiable
5. This then means that the original conclusion was supported by the clauses

Resolution Example

Review: Quick Resolution Example

$$\forall x\ 433Inst(x) \rightarrow Cool(x)$$
$$433Inst(Jon)$$

is

$$Cool(Jon)?$$

UNIVERSITY OF CALGARY

Review: Quick Resolution Example

$$433Inst(x) \rightarrow Cool(x)$$
$$433Inst(Jon)$$

is (we can drop the for all x)

$$Cool(Jon)?$$

Review: Quick Resolution Example

$$433Inst(x) \rightarrow Cool(x)$$
$$433Inst(Jon)$$

is $Cool(Jon)?$

Negate $Cool(Jon)$ to $\neg Cool(Jon)$

Make conjuctive clause combination

$$433Inst(x) \rightarrow Cool(x) \;\wedge\; 433Inst(Jon) \;\wedge\; \neg Cool(Jon)$$

Review: Quick Resolution Example

$$433Inst(x) \rightarrow Cool(x) \;\; \wedge \;\; 433Inst(Jon) \;\; \wedge \;\; \neg\, Cool(Jon)$$

We need CNF (Conjuctive Normal Form)

$$\big(\neg 433Inst(x) \vee Cool(x)\big) \;\; \wedge \;\; \big(433Inst(Jon)\big) \;\; \wedge \;\; (\neg\, Cool(Jon))$$

Review: Quick Resolution Example

$$\left(\neg 433Inst(x) \vee Cool(x)\right) \;\wedge\; \left(433Inst(Jon)\right) \;\wedge\; (\neg\, Cool(Jon))$$

If we have $\left(\neg 433Inst(x) \vee Cool(x)\right)$

And $\left(433Inst(Jon)\right)$

Then using $mgu(x \rightarrow Jon)$ we get $\left(\neg 433Inst(Jon) \vee Cool(Jon)\right)$

We have truth of $433Inst(Jon)$ so for $\left(\neg 433Inst(Jon) \vee Cool(Jon)\right)$ to be true then we must have $\boldsymbol{Cool(Jon)}$ as truth

Now we have knowledge

$$\left(\neg 433Inst(x) \vee Cool(x)\right) \wedge \left(433Inst(Jon)\right) \wedge (\neg\, Cool(Jon)) \wedge (\,\boldsymbol{Cool(Jon)})$$

Review: Quick Resolution Example

$\left(\neg 433Inst(x) \vee Cool(x)\right) \wedge \left(433Inst(Jon)\right) \wedge (\neg\, Cool(Jon)) \wedge (\, Cool(Jon))$

We have a contradiction

$Cool(Jon) \wedge (\neg Cool(Jon))$ resolve to ■

Therefore, the CNF form was unsatisfiable which means the original clauses agree with $Cool(Jon)$

Set-Based Search Applied to Resolution

Facts?

Concrete Example: Resolution (I)

- We describe our world by a collection of special logical formulas, so-called clauses:

$$L_1\left(t_{1,1}, \ldots, t_{1,n1}\right) \vee \cdots \vee L_m\left(t_{m,1}, \ldots, t_{m,nm}\right)$$

where L_i predicate symbol or its negation, $t_{i,j}$ terms out of function symbols and variables (x,y…) variables in different clauses are disjunct

- Examples: $p \vee \neg q, P(a,b,x) \vee R(x,y,c), Q\big(f(a,b), g(x,y)\big), \neg Q(a,b)$

- A consequence we want to prove is negated, transformed into clauses and these clauses are added to the world.
- The consequence is proven, if the empty clause (■) can be deduced.

Extension Rules?

Concrete Example: Resolution (II)

- We derive new clauses by either Resolution or Factorization

Resolution:

$$\frac{C \lor P\ , D \lor \neg P'}{\sigma(C \lor D)} \qquad \text{if } \sigma = \text{mgu}(P, P')$$

mgu = most general unifier

Factorization:

$$\frac{C \lor P \lor P'}{\sigma(C \lor P)} \qquad \text{if } \sigma = \text{mgu}(P, P')$$

Needed: Unification to compute mgu

Yet another set-based search problem:

Concrete Example: Resolution (II)

if $\sigma = \mathrm{mgu}(P, P')$

Examples of P, P’

P(x,y) and P(a,b)

P(a) and P(a)

P(x) and P(b)

Not examples of P, P’ (the predicate letters are different)

P(a) and R(a)

P(a,b) and S(a,b)

R(x) and S(t)

Set-Based Search Applied to Unification

Facts?

Concrete Example: Resolution: Unification (I)

States:

set of term equations $u \approx v$

with $\bot$ (symbol for False) indicating failure

UNIVERSITY OF CALGARY

Extension Rules?

Concrete Example: Resolution: Unification (I)

Extension rules:

Delete:

$$\frac{E \cup \{t \approx t\}}{E}$$

No longer need to maintain a unifier of something to itself

UNIVERSITY OF CALGARY

Concrete Example: Resolution: Unification (II)

Extension rules:

Decompose:

$$\frac{E \cup \{f(t_1, \dots, t_n) \approx f(s_1, \dots, s_n)\}}{E \cup \{t_1 \approx s_1, \dots t_n \approx s_n\}}$$

If you have function unified to same name function, can recompose unifier to only be unifying the internals

UNIVERSITY OF CALGARY

Concrete Example: Resolution: Unification (III)

Extension rules:

Orient:

$$\frac{E \cup \{t \approx x\}}{E \cup \{x \approx t\}}$$

t is not variable

Order of unifier can be changed

Concrete Example: Resolution: Unification (IV)

Extension rules:

Substitute:

$$\frac{E \cup \{x \approx t, t' \approx s'\}}{E \cup \{x \approx t, t'[x \leftarrow t] \approx s'[x \leftarrow t]\}}$$

Can modify one unifier with another as long as x not in t

Concrete Example: Resolution: Unification (V)

Extension rules:

Occurs check:

$$\frac{E \cup \{x \approx t\}}{\bot}$$

If x is in t we cannot unify them (think infinite expansion as issue)

Concrete Example: Resolution: Unification (VI)

Extension rules:

Clash:

$$\frac{E \cup \{f(t_1, \dots, t_n) \approx g(s_1, \dots, s_n)\}}{\bot}$$

If $f \neq g$ we cannot unify them

Constants a $\approx b$ are the same as a$() \approx b()$ and would fall under this rule

Together?

Concrete Example: Resolution: Unification (VII)

Needed: Unification to compute mgu

Yet another set-based search problem:

States:

set of term equations $u \approx v$, with $\bot$ indicating failure

Extension rules:

Delete, Decompose, Orient, Substitute, Occurs check, Clash

Goal condition:

all equations in the state have form
$x \approx t$ and Occurcheck and Substitute are not applicable

Unification/Resolution: Examples

Concrete Example: Resolution (III)

x,y,z are variables, rest are literals, functions, and predicates

Examples for Unification:

(1) $f(g(x,y),c) \approx f(g(f(d,x),z),c)$

(2) $h(c,d,g(x,y)) \approx h(z,d,g(g(a,y),z))$

Examples for Resolution:

(1) $p \lor q, p \lor \neg q, \neg p \lor q, \neg p \lor \neg q$

(2) $P(x) \lor R(x), \neg R(f(a,b)), \neg P(g(a,b))$

(3) $P(x) \lor R(y), \neg R(f(a,b)), \neg P(g(a,b))$

UNIVERSITY OF CALGARY

Concrete Example: Resolution (III)

x,y,z are variables, rest are literals, functions, and predicates

Examples for Unification:

$\{\boldsymbol{f}(g(x,y),c) \approx \boldsymbol{f}(g(f(d,x),z),c)\}$

decompose

$\{g(x,y) \approx g(f(d,x),z),\ \boldsymbol{c} \approx \boldsymbol{c}\}$

delete

$\{\boldsymbol{g}(x,y) \approx \boldsymbol{g}(f(d,x),z)\}$

decompose

$\{\boldsymbol{x} \approx \boldsymbol{f}(\boldsymbol{d},\boldsymbol{x}),\ y \approx z\}$

occurs check ⊥

Concrete Example: Resolution (III)

x,y,z are variables, rest are literals, functions, and predicates

Examples for Unification:

$\{\boldsymbol{h}(c, d, g(x, y) \approx \boldsymbol{h}(z, d, g(g(a, y), z))\}$

decompose

$\{c \approx z, \boldsymbol{d} \approx \boldsymbol{d}, g(x, y) \approx g(g(a, y), z)\}$

delete

$\{\boldsymbol{c} \approx \boldsymbol{z}, g(x, y) \approx g(g(a, y), z)\}$

orient

$\{z \approx c, \boldsymbol{g}(x, y) \approx \boldsymbol{g}(g(a, y), z)\}$

Concrete Example: Resolution (III)

$\{z \approx c, \boldsymbol{g}(x, y) \approx \boldsymbol{g}(g(a, y), z)\}$

decompose

$\{z \approx c, x \approx g(a, y), \boldsymbol{y} \approx \boldsymbol{z}\}$

substitute

$\{z \approx c, \boldsymbol{x} \approx \boldsymbol{g}(\boldsymbol{a}, \boldsymbol{y}), y \approx c\}$

substitute

$\{z \approx c, x \approx g(a, c), y \approx c\}$

done

Concrete Example: Resolution (III)

x,y,z are variables, rest are literals, functions, and predicates

Examples for Unification:

(1) $f(g(x,y),c) \approx f(g(f(d,x),z),c)$ occur check ⊥

(2) $h(c,d,g(x,y)) \approx h(z,d,g(g(a,y),z))$ mgu = $\{z \approx c, x \approx g(a,c), y \approx c\}$

Examples for Resolution:

(1) $p \lor q, p \lor \neg q, \neg p \lor q, \neg p \lor \neg q$

(2) $P(x) \lor R(x), \neg R(f(a,b)), \neg P(g(a,b))$

(3) $P(x) \lor R(y), \neg R(f(a,b)), \neg P(g(a,b))$

Concrete Example: Resolution (III)

(1) $p \vee q$

(2) $p \vee \neg q$

(3) $\neg p \vee q$

(4) $\neg p \vee \neg q$

(5) $p \vee p$ resolve (1) and (2)

(6) p factorize (5)

(7) $\neg p \ \vee \neg p$ resolve (3) and (4)

(8) $\neg p$ factorize (7)

(9) ■ resolving (6) and (8)

Resolution:

$$\frac{C \vee P\ ,D \vee \neg P'}{\sigma(C \vee D)}$$

Factorization:

$$\frac{C \vee P \vee P'}{\sigma(C \vee P)}$$

Concrete Example: Resolution (III)

x,y,z are variables, rest are literals, functions, and predicates

Examples for Resolution:

(1) $P(x) \vee R(x)$

(2) $\neg R(f(a,b))$

(3) $\neg P(g(a,b))$

(4) $P(f(a,b))$ resolving (1) and (2) with $mgu = \{x \approx f(a,b)\}$

(5) $R(g(a,b))$ resolving (1) and (3) with $mgu = \{x \approx g(a,b)\}$

Can't reach empty clause

Resolution:

$$\frac{C \vee P\,,\, D \vee \neg P'}{\sigma(C \vee D)}$$

Factorization:

$$\frac{C \vee P \vee P'}{\sigma(C \vee P)}$$

Concrete Example: Resolution (III)

x,y,z are variables, rest are literals, functions, and predicates

Examples for Resolution:

(1) $P(x) \lor R(y)$

(2) $\neg R(f(a,b))$

(3) $\neg P(g(a,b))$

(4) $\mathrm{P}(x)$ resolving (1) and (2) with $mgu = \{y \approx f(a,b)\}$

(5) ■ resolving (3) and (4) with $mgu = \{x \approx g(a,b)\}$

Resolution:

$$\frac{C \lor P\,,\, D \lor \neg P'}{\sigma(C \lor D)}$$

Factorization:

$$\frac{C \lor P \lor P'}{\sigma(C \lor P)}$$

UNIVERSITY OF CALGARY

Concrete Example: Resolution (III)

x,y,z are variables, rest are literals, functions, and predicates

Examples for Unification:

(1) $f(g(x,y),c) \approx f(g(f(d,x),z),c)$ occur check ⊥

(2) $h(c,d,g(x,y)) \approx h(z,d,g(g(a,y),z))$ mgu = $\{z \approx c, x \approx g(a,c), y \approx c\}$

Examples for Resolution:

(1) $p \lor q, p \lor \neg q, \neg p \lor q, \neg p \lor \neg q$ produced empty clause

(2) $P(x) \lor R(x), \neg R(f(a,b)), \neg P(g(a,b))$ couldn't reach empty clause

(3) $P(x) \lor R(y), \neg R(f(a,b)), \neg P(g(a,b))$ produced empty clause

Unification/Resolution: Set-Based

Concrete Example: Resolution (V)

Tasks:

- Describe Resolution as set-based search model
  - $\boldsymbol{F}, \boldsymbol{Ext}$
- Given the following control idea, describe formally a search control for your model, so that we have a search process:
  - $\boldsymbol{f_{wert}}, \boldsymbol{f_{select}}$

Perform factorization whenever possible; choose the smallest possible clauses for resolution; if several clause pairs are smallest, use an ordering $<_{\text{Lit}}$ on the predicates and terms

Concrete Example: Resolution (VI) Model

$$\boldsymbol{F} = \{f_1, \dots, f_t\}$$

Concrete Example: Resolution (VI) Model

$$\boldsymbol{F} = \{f_1, \dots, f_t \mid f_i = \mathrm{L}_1(t_{1,1}, \dots, t_{1,n1}) \vee \cdots \vee L_m(t_{m,1}, \dots, t_{m,nm})\}$$

set of t facts

where each fact is formed where L_i predicate symbol or its negation

$t_{i,j}$ terms out of function symbols and variables (x,y…) variables in different clauses are disjunct}

Concrete Example: Resolution (VI) Model

$$\boldsymbol{Ext} = \{A \rightarrow B \mid A, B \subseteq \boldsymbol{F} \; and \; \}$$

Concrete Example: Resolution (VI) Model

$$\boldsymbol{Ext} = \{A \rightarrow B \mid A, B \subseteq \boldsymbol{F} \text{ } and \text{ } Resolution(A, B), Factorization(A, B)\}$$

Concrete Example: Resolution (VI) Model

$$\boldsymbol{Ext} = \{A \rightarrow B \mid A, B \subseteq \boldsymbol{F} \text{ and } Resolution(A, B), Factorization(A, B)\}$$

$$Resolution(A, B) = \frac{C}{D} \text{ where } A = C \text{ and } B = C \cup D$$

$$Factorization(A, B) = \frac{C}{D} \text{ where } A = C \text{ and } B = C \cup D$$

Resolution:

$$\frac{C \vee P\ , D \vee \neg P'}{\sigma(C \vee D)}$$

Factorization:

$$\frac{C \vee P \vee P'}{\sigma(C \vee P)}$$

UNIVERSITY OF CALGARY

Concrete Example: Resolution (VI) Model

$$\boldsymbol{Ext} = \{A \rightarrow B \mid A, B \subseteq \boldsymbol{F} \text{ and } Resolution(A, B), Factorization(A, B)\}$$

Resolution:

$$\frac{C \vee P \ , D \vee \neg P'}{\sigma(C \vee D)}$$

$$Resolution(A, B) = \frac{E}{F} \qquad where\ A = E\ and\ B = E \cup F$$

Produce a new clause F from clauses in E

Factorization:

$$\frac{C \vee P \vee P'}{\sigma(C \vee P)}$$

$$Factorization(A, B) = \frac{E}{F} \qquad where\ A = E\ and\ B = E \cup F$$

Produce a new clause E from clauses in E

Concrete Example: Resolution (VI) Process

- $f_{wert}(A, B, e) = \mathbb{N}$
  - If $A \rightarrow B$ exists that fulfils $Factorization(A, B) = \frac{E}{F}$ with $E \notin s$ then $f_{wert}(A, B, e) = 0$ (always choose factorization)
  - if $A \rightarrow B$ exists that fulfills $Resolution(A, B) = \frac{E}{F}$ with $E \notin s$ then $f_{wert}(A, B, e) = size(A)$ where $size(A)$ is a summation of size of clauses in A (next do Resolution based on size)

Concrete Example: Resolution (VI) Process

- $f_{wert}(A, B, e) = \mathbb{N}$
  - always choose Factorization first
  - next do Resolution based on size

Concrete Example: Resolution (VI) Process

- $f_{select}(\{A' \to B'\}, e) = A \to B$
  - where $A \to B$ is at index 0 after creating a sorted order of $\{A' \to B'\}$ according to ordering $<_{\text{Lit}}$
  - there should exists no two clauses which cannot be ordered by $<_{\text{Lit}}$ as there are no duplicates

Concrete Example: Resolution (VI) Process

- $f_{select}(\{A' \rightarrow B'\}, e) = A \rightarrow B$
  - use ordering for tie break

Unification/Resolution: Set-Based: Applied

Concrete Example: Resolution (VI)

Tasks (cont.):

- Apply your process to the search instance to the following set of clauses:

$$\left\{\begin{matrix} \neg P(x,y) \vee P(y,x), \\ P\big(f(x),g(y)\big) \vee \neg R(y), \\ \neg P\big(g(x),f(x)\big), \\ R(x) \vee Q(x,b), \\ \neg Q(a,x) \end{matrix}\right\}$$

Concrete Example: Resolution (VI)

Tasks (cont.):

- Remember its best to think of variables in each clause as independent variables

$$\left\{\begin{matrix} \neg P(x_1, y_1) \vee P(y_1, x_1), \\ P\big(f(x_2), g(y_2)\big) \vee \neg R(y_2), \\ \neg P\big(g(x_3), f(x_3)\big), \\ R(x_4) \vee Q(x_4, b), \\ \neg Q(a, x_5) \end{matrix}\right\}$$

Concrete Example: Resolution (VI)

Tasks (cont.):

- Last two resolved

$$\left\{\begin{array}{c} \neg P(x_1, y_1) \lor P(y_1, x_1), \\ P\big(f(x_2), g(y_2)\big) \lor \neg R(y_2), \\ \neg P\big(g(x_3), f(x_3)\big), \\ \boldsymbol{R(x_4)} \lor \boldsymbol{Q(x_4, b)}, \\ \neg \boldsymbol{Q(a, x_5)}, \\ \boldsymbol{R(a)} \end{array}\right\}$$

mgu = $\{x_4 \approx a, x_5 \approx b\}$

UNIVERSITY OF CALGARY

Concrete Example: Resolution (VI)

Tasks (cont.):

- Resolve newest with 2nd

$$\left\{\begin{array}{c} \neg P(x_1, y_1) \lor P(y_1, x_1), \\ \boldsymbol{P(f(x_2), g(y_2)) \lor \neg R(y_2)}, \\ \neg P(g(x_3), f(x_3)), \\ R(x_4) \lor Q(x_4, b), \\ \neg Q(a, x_5), \\ \boldsymbol{R(a)}, \\ \boldsymbol{P(f(x_2), g(a))} \end{array}\right\}$$

mgu = $\{y_2 \approx a\}$

Concrete Example: Resolution (VI)

Tasks (cont.):

- Resolve newest with first

$$\left\{\begin{array}{c} \neg \boldsymbol{P}(\boldsymbol{x_1}, \boldsymbol{y_1}) \lor \boldsymbol{P}(\boldsymbol{y_1}, \boldsymbol{x_1}), \\ P\big(f(x_2), g(y_2)\big) \lor \neg R(y_2), \\ \neg P\big(g(x_3), f(x_3)\big), \\ R(x_4) \lor Q(x_4, b), \\ \neg Q(a, x_5), \\ R(a), \\ \boldsymbol{P}\big(\boldsymbol{f}(\boldsymbol{x_2}), \boldsymbol{g}(\boldsymbol{a})\big) \\ \boldsymbol{P}(\boldsymbol{g}(\boldsymbol{a}), \boldsymbol{f}(\boldsymbol{x_2})) \end{array}\right\}$$

mgu = $\{x_1 \approx f(x_2), y_1 \approx g(a)\}$

Concrete Example: Resolution (VI)

Tasks (cont.):

- Resolve newest with third

$$\left\{\begin{array}{c} \neg P(x_1, y_1) \lor P(y_1, x_1), \\ P\big(f(x_2), g(y_2)\big) \lor \neg R(y_2), \\ \boldsymbol{\neg P\big(g(x_3), f(x_3)\big)}, \\ R(x_4) \lor Q(x_4, b), \\ \neg Q(a, x_5), \\ R(a), \\ P\big(f(x_2), g(a)\big) \\ \boldsymbol{P(g(a), f(x_2))} \\ \blacksquare \end{array}\right\}$$

mgu = $\{x_3 \approx a, x_2 \approx a\}$

UNIVERSITY OF CALGARY

Onward to …
genetic algorithm set-based search

Jonathan Hudson, Ph.D.
jwhudson@ucalgary.ca
https://cspages.ucalgary.ca/~jwhudson/

UNIVERSITY OF CALGARY