Logic Systems

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

Logics

Logics

- Considered by humans as the knowledge representation (and processing) method of computers
- Clear mathematical foundation:
  syntax describes formulas; axioms what is considered true; inference rules how to get other true formulas
- Many different kinds of logics
- Meaning of a formula usually not easy to determine by humans (rather formal semantics)

General Definitions (I)

Syntax:

Terms (without sorts): $\boldsymbol{F} = F$ (function symbols) $\cup$ V (function variables);

$\tau(f) \in \mathbb{N}$ multiplicity where $f \in Term$

$Term(\boldsymbol{F})$ recursively defined by

if $f \in \boldsymbol{F}$ with $\tau(f) = n$ and $t_1, \ldots, t_n \in Term(\boldsymbol{F})$

then $f(t_1, \ldots, t_n) \in Term(\boldsymbol{F})$

General Definitions (I)

Example function symbols f,g,h   (but also a,b,c)

Example function variables x,y,z

f(x), g(x,y), h(x,y,z)

f(a), g(b,y), h(x,c,d)

Also note,

Note a() = a, b()=b, c() = c

General Definitions (I)

Syntax:

Atoms: $\mathbf{P} = P$ (predicate symbols) ∪ PI (interpreted predicate symbols) $\cup$ PV (predicate variables);

$\tau(A) \in \mathbb{N}$ multiplicity where $A \in Atom$

$$Atom = Atom\big(\boldsymbol{P}, Term(\boldsymbol{F})\big)$$

$$= \{A(t_1, \dots, t_n) \mid A \in \boldsymbol{P}, \tau(A) = n, t_1, \dots, t_n \in Term(\boldsymbol{F})\}$$

General Definitions (I)

Example Atoms:

Predicate Symbols P,Q,R

Inter. Predicate Symbols EQ (equality)

Predicate Variables X,Y,Z

P(x), Q(x,y), R(f(x),z)

EQ(x,y)

X(x), Y(a), Z(x,c,f(a))

General Definitions (II)

Formulas: sets J (Junctors), Q (Quantifiers);

$\tau(\star) \in \mathbb{N}$ multiplicity where $\star \in J$

$\tau(\square) \in \mathbb{N}$ multiplicity where $\square \in Q$

$Form = Form\big(J, Q, Atom(\boldsymbol{P}, Term(\boldsymbol{F})\,)\big)$

recursively def.

- $A \in Form$ if $A \in Atom$
- $\star \in J, \tau(\star) = n, A_1, \dots, A_n \in Form$ then $\star\,(A_1, \dots, A_n) \in Form$
- $\square \in Q, A \in Form, x_1, \dots, x_n \in V \cup PV$ then $\square x_1, \dots, x_n . A \in Form$

General Definitions (II)

Formulas: sets J (Junctors), Q (Quantifiers);

$\tau(\star) \in \mathbb{N}$ multiplicity   where $\star \in J$

$\tau(\square) \in \mathbb{N}$ multiplicity   where $\square \in Q$

Example Junctors

$\wedge, \vee, \neg, \rightarrow, \leftrightarrow$

Example Quantifiers

$\forall, \exists$

$$\forall x. \exists y. P(x, y) \vee Q(x) \wedge EQ(f(x), y)$$

UNIVERSITY OF CALGARY

General Definitions (III)

Adding Meaning:

Interpretation: Given $Form\left(J, Q, Atom\left(\boldsymbol{P}, Term(\boldsymbol{F})\right)\right)$, set D of objects (domain), set W of truth values

Interpretation I

- Assigns to each $f \in \boldsymbol{F}$ a function over D and to each $A \in \boldsymbol{P}$ a predicate over D in the truth values of W
- Assigns to each $\star \in J, \tau(\star) = n$, a function $W^n \rightarrow W$
- Assigns to each $\square \in Q$ a combination rule for truth values in W, such that $I(x_1, \ldots, x_n . B)$ is determined by combining the truth values of all the formulas that are generated by substituting the variables $x_1, \ldots, x_n$ in B by arbitrary (but fitting) combinations of functions and/or predicates over D

General Definitions (IV)

All together:

Logic: $Form, I = \{I_1, I_2, \ldots\}$ a set of interpretations with

- $I_i(\star) = I_j(\star)\ \forall i, j$ and $\star \in J$
- $I_i(\square) = I_j(\square)\ \forall i, j$ and $\square \in Q$
- $I_i(A) = I_j(A)\ \forall i, j$ and $A \in PI$ (interpreted predicates)

☞ $(Form, I)$ logic

Note: there are many different logics!

Working with a Logic

Calculus:

$(Form, I)$ logic to W. $Ax \subseteq Form$ set of Axioms; R set of rules:
  (Ax, R) calculus to $(Form, I)$ and $w \in W$, if
    $B \in Form$ with $\mathrm{I}(B) = w$ for all $\mathrm{I} \in I$ can be transformed into subset of Ax by applying the rules of R

Note: this still allows for different search models using the calculus rules!

Propositional Logic

Propositional logic

General idea:

- Formulas describe combinations of statements (propositions) that are either truth or false and this way build statements themselves.
- No parameterized statements!
- Basis of the logics of gates, circuits and micro chips

Basic knowledge structures

- $Term(\boldsymbol{F}) = \emptyset$ there are no terms (only predicates)
- $\boldsymbol{P} = P$ and $\tau(A) = 0 \; \forall A \in P$ There are only predicates (not PV or PI) and there are no arguments to any predicate (we often just use lower case for our predicates)
  - (elements of P sometimes called propositional variables; very unfortunate naming!)
- $J = \{\neg, \vee, \wedge, \rightarrow, \leftrightarrow\}, Q = \emptyset$
- $W = \{\text{true}, \text{false}\}$
- I = all possible interpretations
  (Interpretation here is an assignment of truth values to the propositions in P)

UNIVERSITY OF CALGARY

Semantics

- Look for tautologies, i.e. formulas that are interpreted to true by all $\mathrm{I} \in \mathcal{I}$
- if $I(p) = \text{false then } I(\neg p) = \text{true, otherwise false}$
- if $I(p) \vee I(q) = \text{true then } I(p \vee q) = \text{true, otherwise false}$
- if $I(p) \wedge I(q) = \text{true then } I(p \wedge q) = \text{true, otherwise false}$
- if $I(p) = true \wedge I(q) = \text{false then } I(p \rightarrow q) = \text{false, otherwise true}$
- if $I(p) = I(q) \text{ then } I(p \leftrightarrow q) = \text{true, otherwise false}$

How to get knowledge into the representation structure

- assign predicate symbols to simple positive statements
- Connect them to form complicated statements
- But be careful: “tertium non datur” (no third possibility is given)
  - The car is green =: p
  - The car is red =: q
  - We need in addition:
    $q \leftrightarrow \neg p$

UNIVERSITY OF CALGARY

Discussion

- Decidable, NP complete
  - Decidable
    - there exists an effective method for deriving the correct answer
  - but NP complete
    - nondeterministic polynomial time complete
    - quick to verify solutions, can be brute forced and can simulate all others in NP-complete class;

– not very expressive

– knowledge bases get very large

UNIVERSITY OF CALGARY

And what about processing data?

- Calculus used in most (best) systems:
  Davis-Putnam (working on clauses; special case of Modelelimination)
- Each formula can be transformed into equivalent set of clauses (remember: formula with $J = \{\neg, \vee\}$)
  - "defining" equations for → and ↔
  - DeMorgan's laws to move negation inward
- For deciding tautologies, we use and-tree-based search
- For testing for satisfiability, we see clauses as constraints and use or-tree-based search

Propositional Logic Example

Example

- Represent the following statements in propositional logic:
  - A Porsche is a black car.
  - Black cars are fast cars.
  - Bad cars are slow cars.
- Home exercise:
  Show that the following statement is a logical consequence of the statements above:
  - A Porsche is a good car.

Example

- Represent the following statements in propositional logic:
  - A Porsche is a black car. $porsche \land black$
  - Black cars are fast cars. $black \rightarrow fast$
  - Bad cars are slow cars. $bad \rightarrow \neg fast$
- Home exercise:
  Show that the following statement is a logical consequence of the statements above:
  - A Porsche is a good car.

UNIVERSITY OF CALGARY

Example

- $porsche \land black$
- $black \rightarrow fast$
- $bad \rightarrow \neg fast$
- A Porsche is a good car.

Example

- $p \wedge bl$
- $bl \rightarrow f$
- $b \rightarrow \neg f$
- A Porsche is a good car.

Example

- $p \wedge bl$
- $bl \rightarrow f$
- $b \rightarrow \neg f$
- $p \wedge \neg b$

UNIVERSITY OF CALGARY

Example

- $p \wedge bl$
- $bl \rightarrow f$
- $b \rightarrow \neg f$
- $\neg(p \wedge \neg b) = \neg p \vee b$

UNIVERSITY OF CALGARY

Example

- $p \land bl$
- $bl \rightarrow f$
- $b \rightarrow \neg f$
- $\neg p \lor b$

Example

- p
- bl
- $bl \rightarrow f$
- $b \rightarrow \neg f$
- $\neg p \lor b$

UNIVERSITY OF CALGARY

Example

- p
- bl
- $\neg bl \lor f$
- $\neg b \lor \neg f$
- $\neg p \lor b$

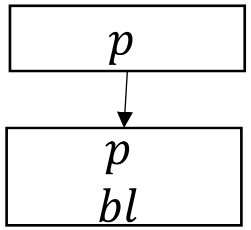
Example

- p
- bl
- $\neg bl \vee f$
- $\neg b \vee \neg f$
- $\neg p \vee b$

$\boxed{p}$

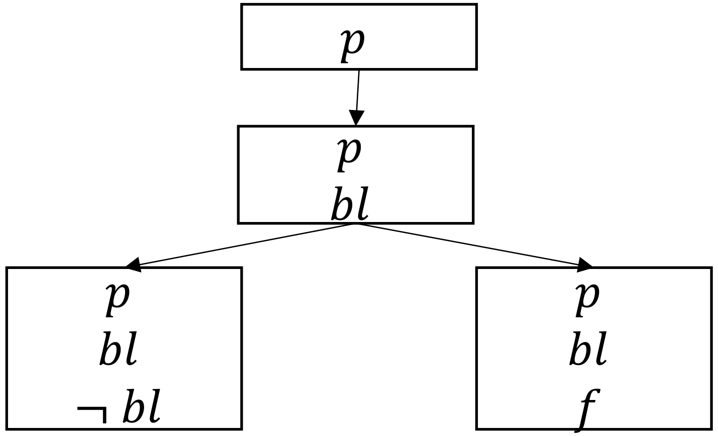
Example

- p
- bl
- $\neg bl \lor f$
- $\neg b \lor \neg f$
- $\neg p \lor b$



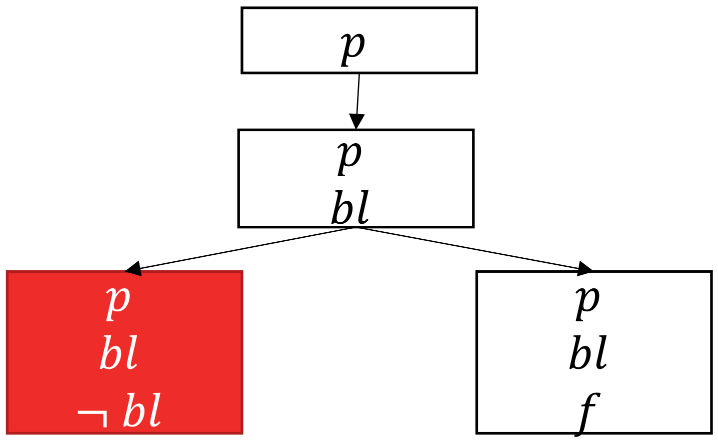
Example

- p
- bl
- $\neg bl \vee f$
- $\neg b \vee \neg f$
- $\neg p \vee b$



Example

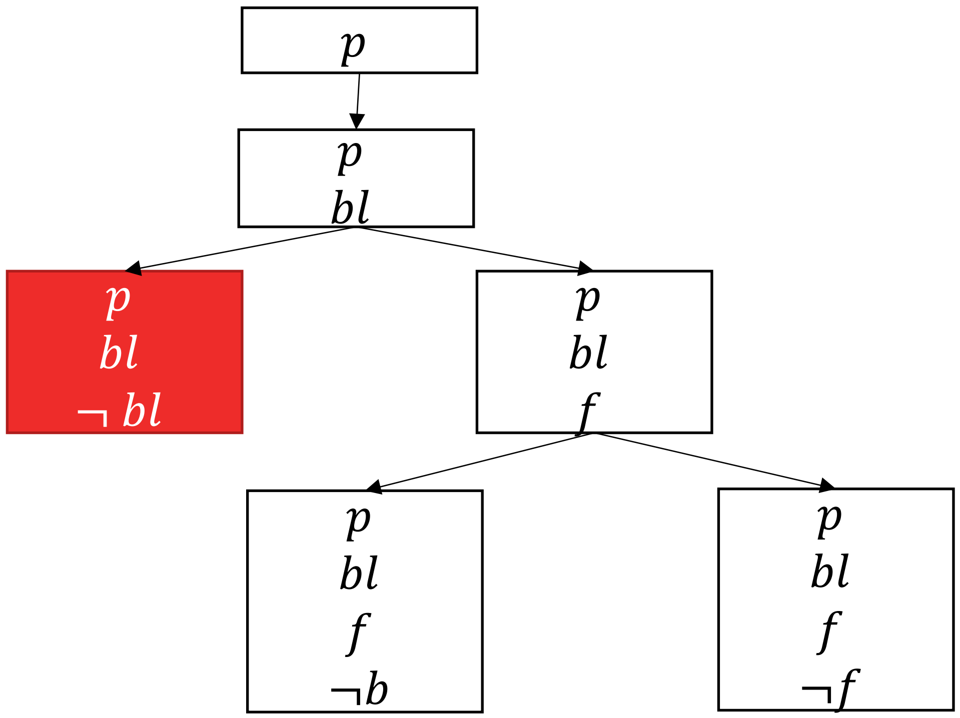
- p
- bl
- $\neg bl \vee f$
- $\neg b \vee \neg f$
- $\neg p \ \vee b$



UNIVERSITY OF CALGARY

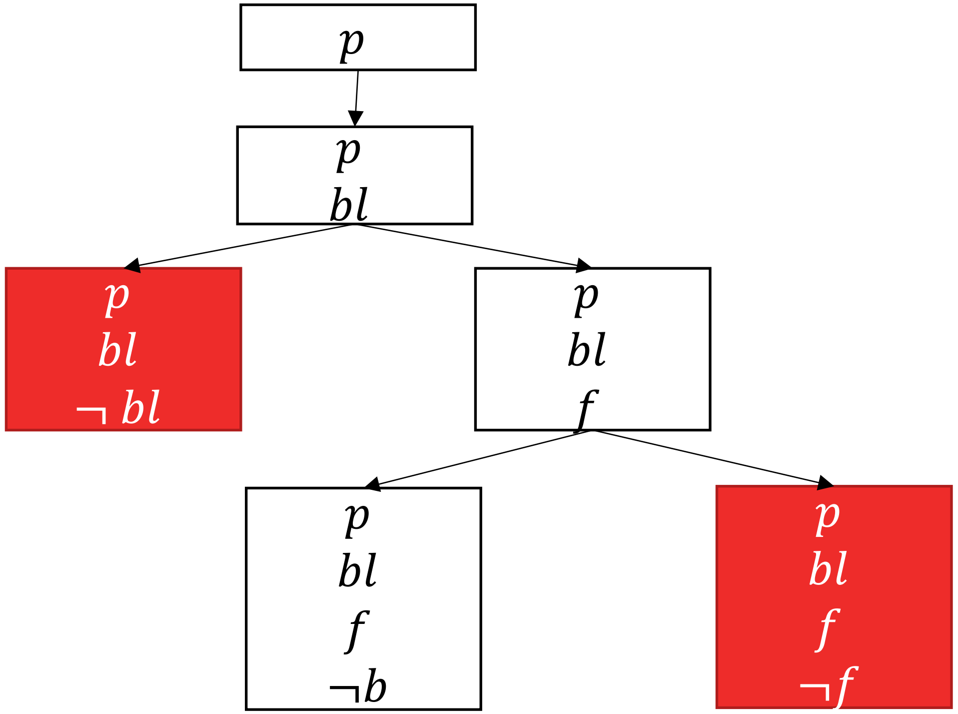
Example

- p
- bl
- $\neg bl \vee f$
- $\neg b \vee \neg f$
- $\neg p \ \vee b$



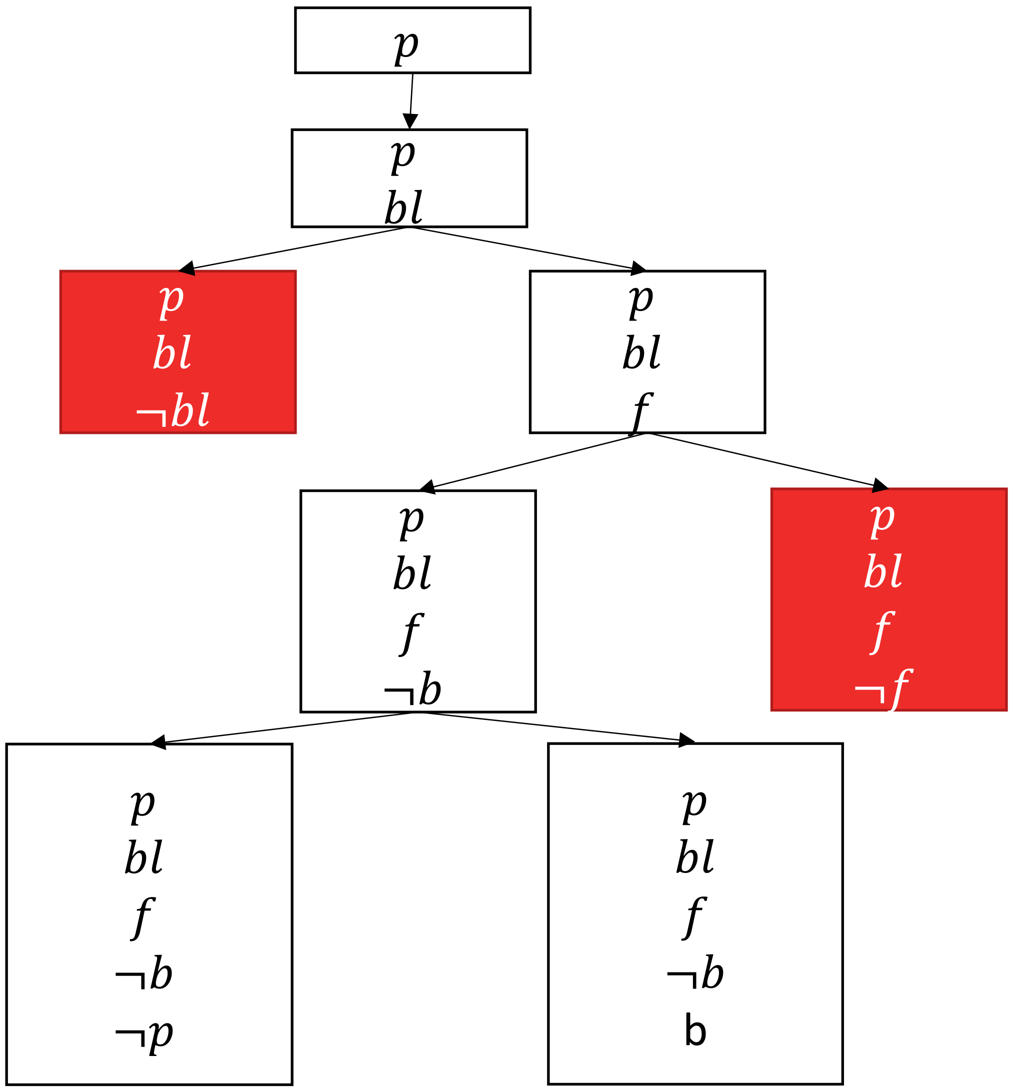
Example

- p
- bl
- $\neg bl \lor f$
- $\neg b \lor \neg f$
- $\neg p \lor b$



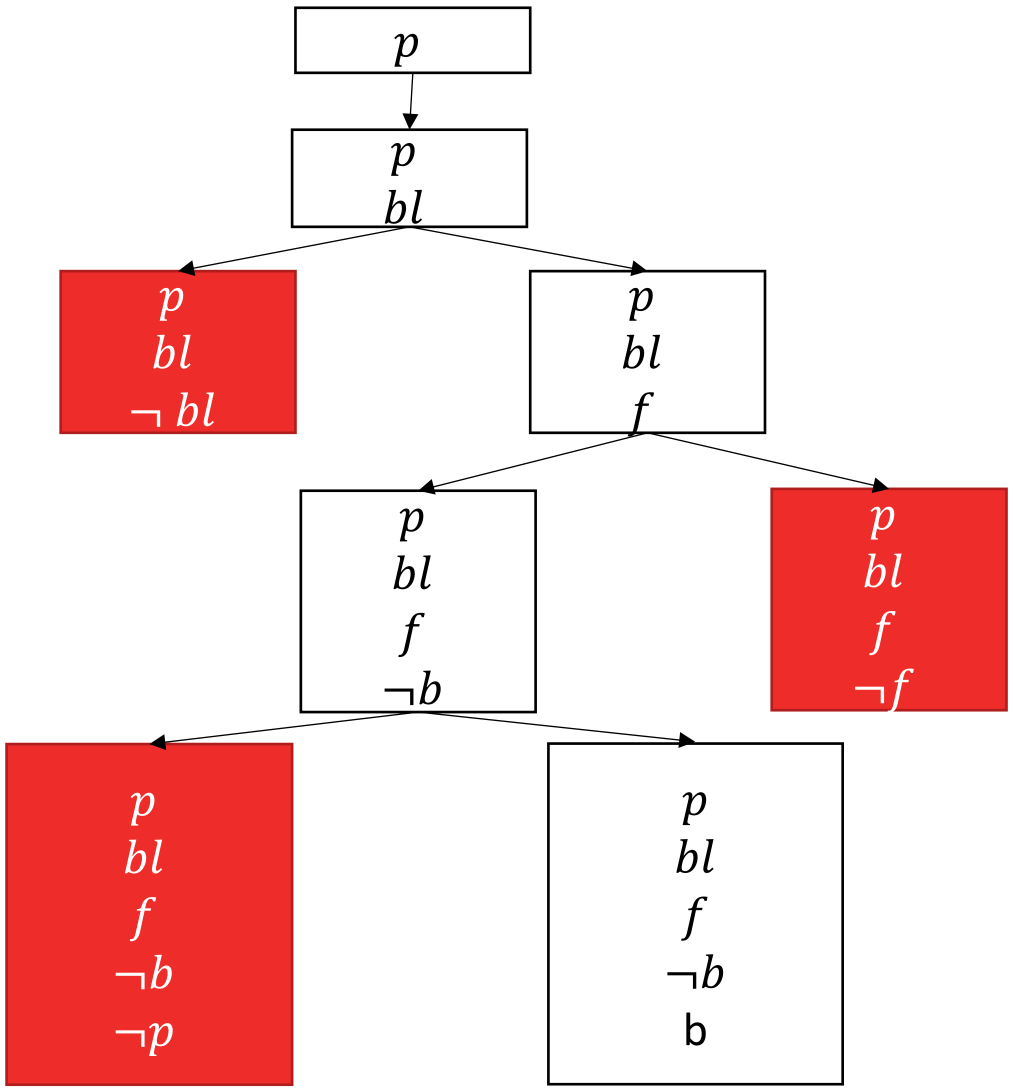
Example

- p
- bl
- $\neg bl \vee f$
- $\neg b \vee \neg f$
- $\neg p \vee b$



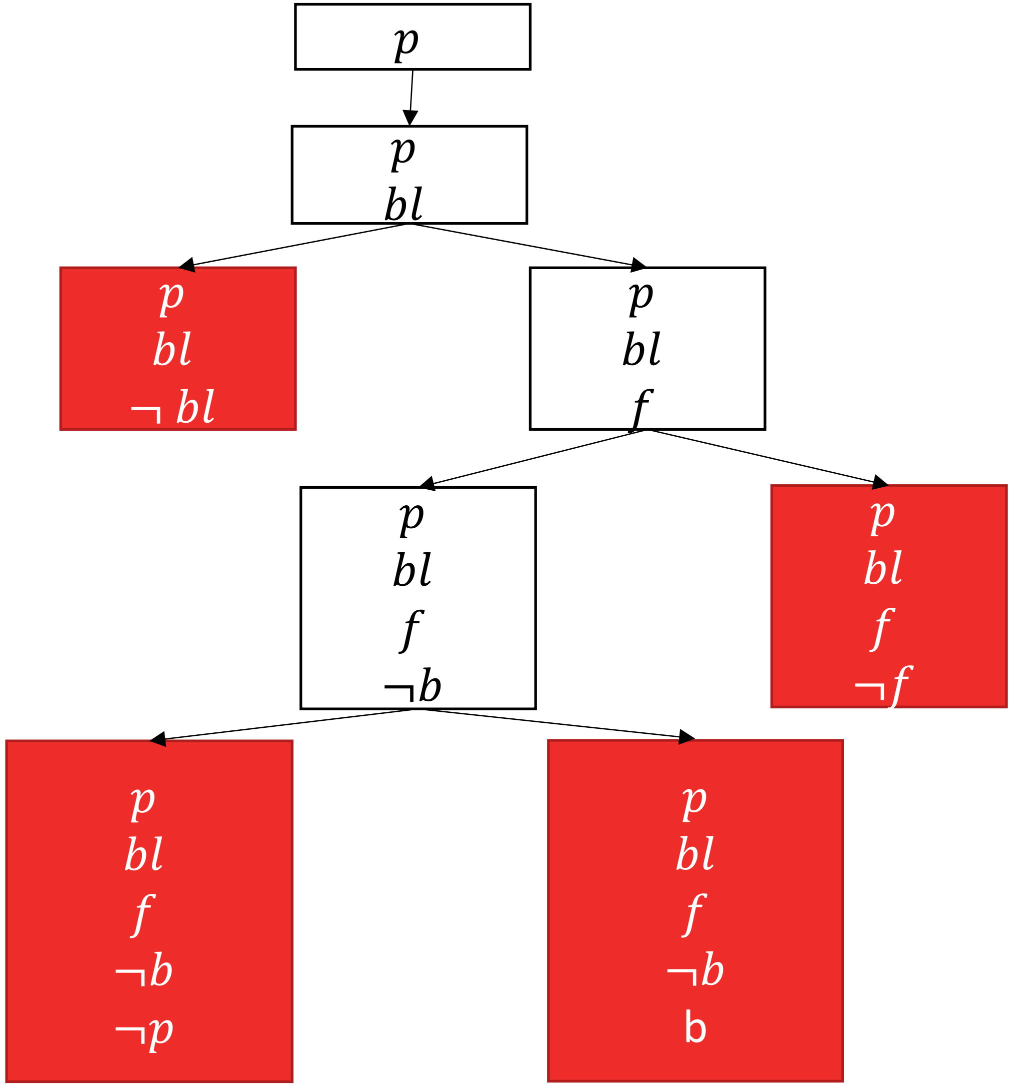
Example

- p
- bl
- $\neg bl \vee f$
- $\neg b \vee \neg f$
- $\neg p \ \vee b$



Example

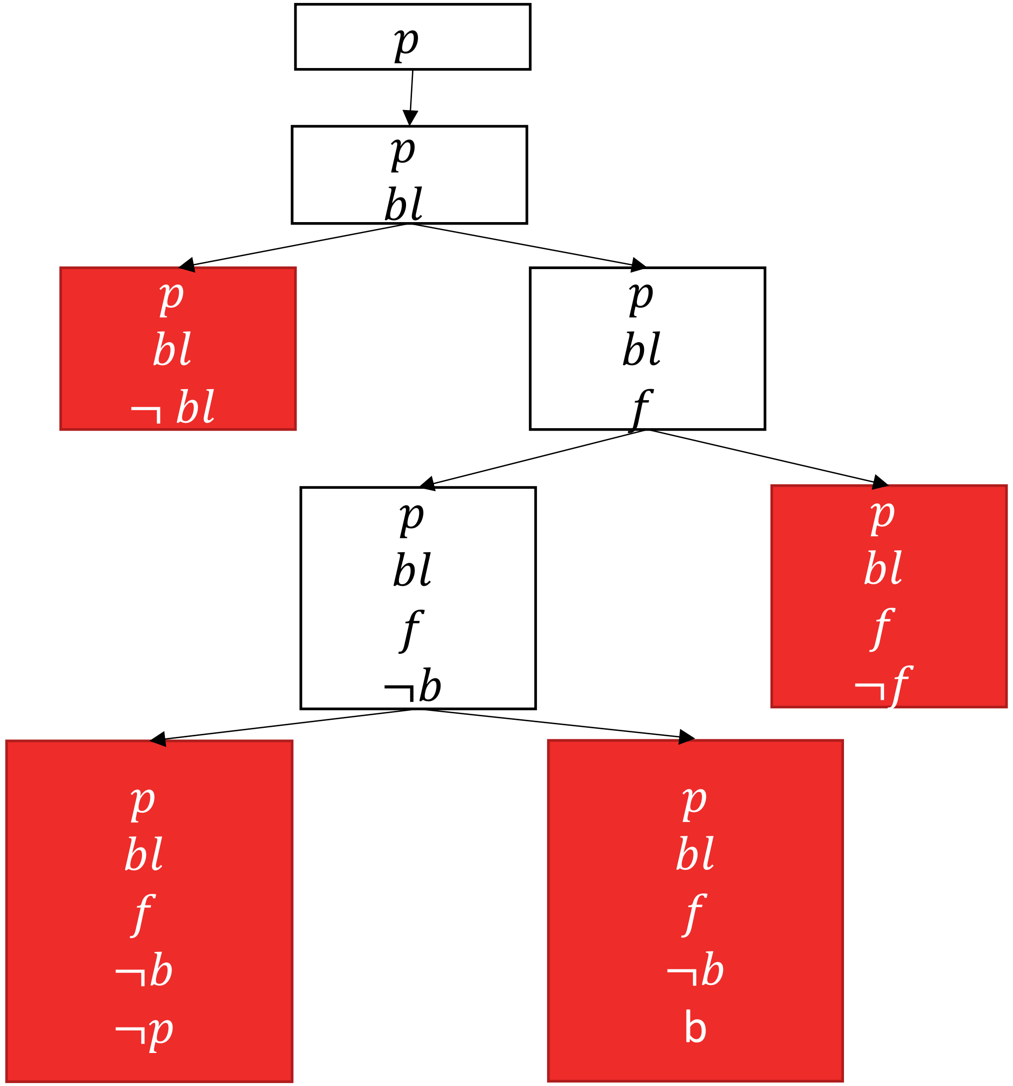
- p
- bl
- $\neg bl \vee f$
- $\neg b \vee \neg f$
- $\neg p \ \vee b$



UNIVERSITY OF CALGARY

Example

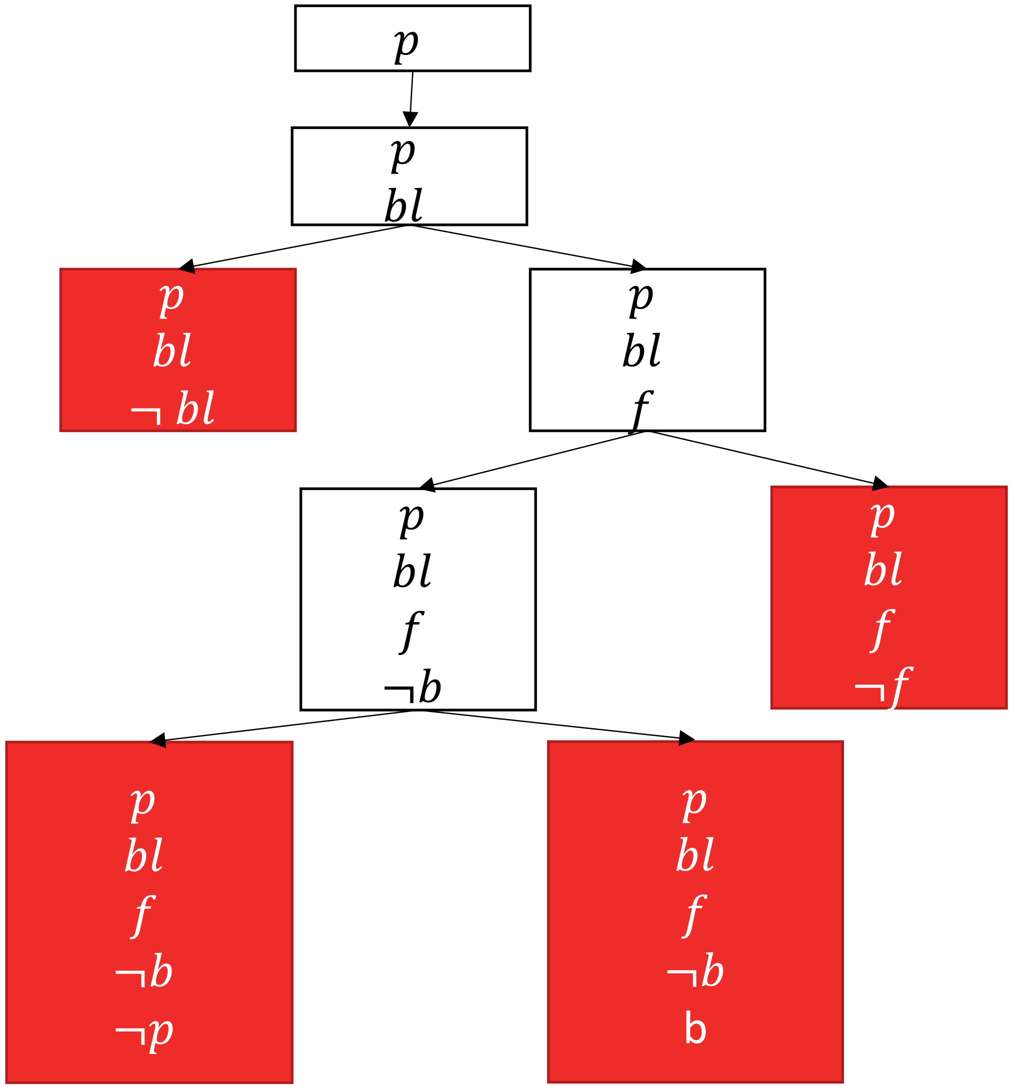
- p
- bl
- $\neg bl \lor f$
- $\neg b \lor \neg f$
- $\neg p \lor b$



Example

- p
- bl
- $\neg bl \lor f$
- $\neg b \lor \neg f$
- $\neg p \lor b$

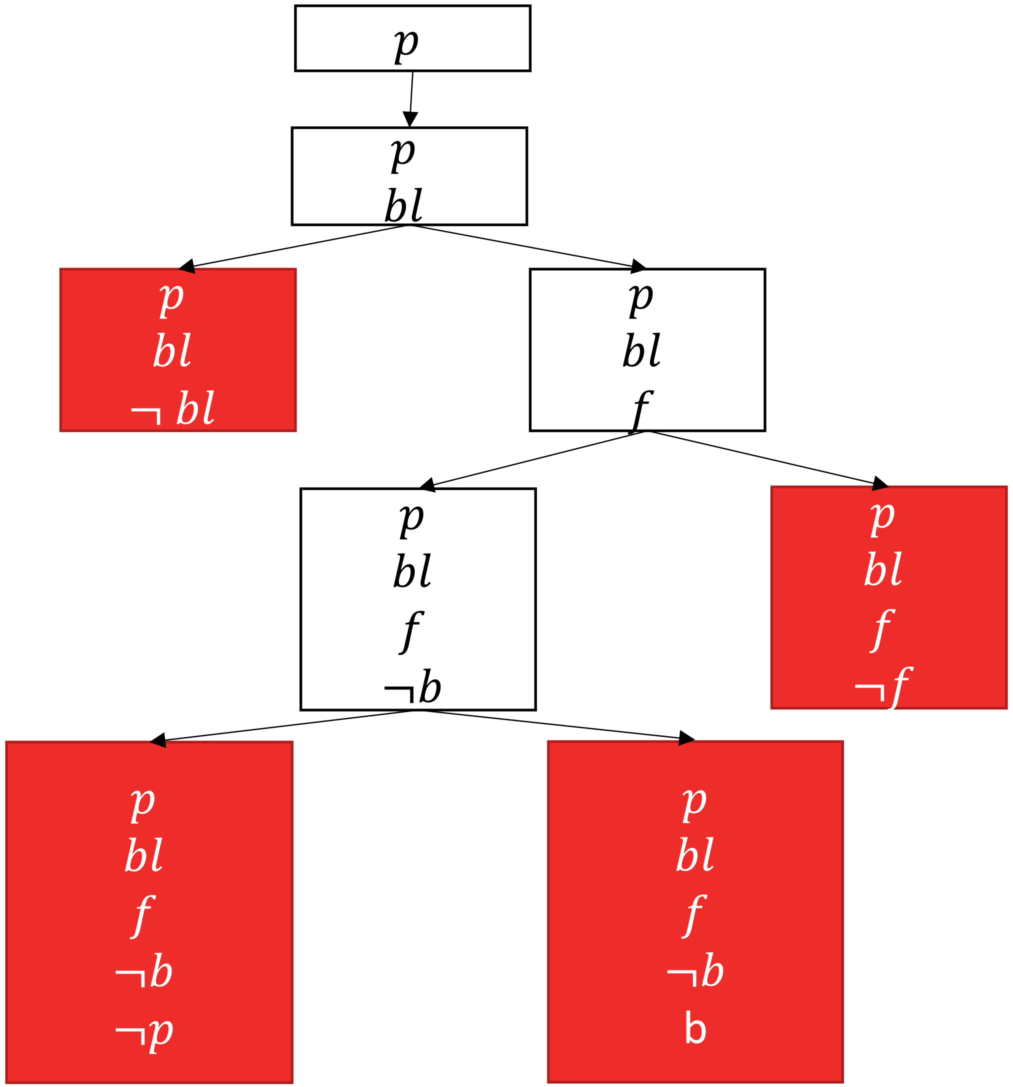
- $\neg(\neg p \lor b)$



Example

- p
- bl
- $\neg bl \lor f$
- $\neg b \lor \neg f$
- $\neg p \lor b$

- $p \land \neg b$
- A Porsche is a good car.



UNIVERSITY OF CALGARY

First-Order Logic

First-order logic

General ideas:

- Introduce terms (complex structures) and data element variables into formulas
  ☞ parameters (formula represents a whole set of
  more specialized ones)
- Talk about the existence of a certain data element and about properties of all possible data elements

Basic knowledge structures

- $Term(\boldsymbol{F})$: F not restricted, $\tau(x) = 0$ f. a. $x \in V$
- $\boldsymbol{P}$: P unrestricted, $PV = \emptyset$, PI depends on what is required (desired)
  Example for predicates in $PI := \{EQ\}$
- $J = \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$, $Q = \{\forall, \exists\}$
- $W = \{\text{true}, \text{false}\}$
- I = all possible (imaginable) interpretations (within the limits given by PI)

Semantics

- Look for tautologies, again.
- Interpret terms and atoms as described earlier.
- Interpret junctors as for propositional logic.
- Let $I^{x,d}(B)$ be the interpretation that assigns to x the data element $d \in D$, i.e. $I^{x,d}(x) = d$.
- if $I^{x,d}(B)$ = true for all $d \in D$ then $I(\forall x.B)$ = true; otherwise false.
- if $I^{x,d}(B)$ = true for one $d \in D$ then $I(\exists x.B)$ = true; otherwise false.
- Quite some freedom for elements of PI (as long as all interpretations agree in it).

How to get knowledge into the representation structure

- Define data objects, functions and predicates you are interested in and map them into terms and atoms.
- Select predicates you want to be treated special ☞ PI
  Note that usually you have then to provide a way to process these special predicates!
- Define all "laws" that you want your objects to obey and make them into formulas, resp. axioms.

Discussion

- Semi-decidable
  - there is a deterministic algorithm such that
    - (a) if an element is a member of the set, the algorithm halts with the result "positive", and
    - (b) if an element is not a member of the set, (i) the algorithm does not halt, or (ii) if it does, then with the result "negative".

- A lot of other logics can be transformed into PL1
  but: formulas are then not easily readable (and understandable) by humans

– Usually all possible interpretations are more than what we really want
  ☞ axioms needed to narrow the true formulas down!

And what about processing data?

- Two types of calculi dominant:
  - Resolution-based (superposition-based)
  - Modelelimination-based

- In both, formula is negated and transformed into set of clauses

- Resolution ☞ set-based search for empty clause
  Modelelimination
  ☞ usually realized with iterative deepening and
  backtracking in and-tree as control

First-Order Logic Example

Example

- Use PL1 for the example for propositional logic (2!)
- Home exercise:
  Show that the statements
  - Everyone who lies is a bad person
  - I know a politician who lies

  implies the statement
  - There is a politician who is a bad person

Example

- Represent the following statements in propositional logic:
  - A Porsche is a black car. black(p)
  - Black cars are fast cars. for all x black(x) -> fast(x)
  - Bad cars are slow cars. for all x bad(x) -> not fast(x)
- Home exercise:
  Show that the following statement is a logical consequence of the statements above:
  - A Porsche is a good car. good(p)

Example

- Represent the following statements in propositional logic:
  - A Porsche is a black car. black(p)
  - Black cars are fast cars. for all x black(x) -> fast(x)
  - Bad cars are slow cars. for all x bad(x) -> not fast(x)
- Home exercise:
  Show that the following statement is a logical consequence of the statements above:
  - A Porsche is a good car. not bad(p)

Example

- black(p)
- for all x black(x) -> fast(x)
- for all x bad(x) -> not fast(x)
- not bad(p)

Example

- black(p)
- for all x black(x) -> fast(x)
- for all x bad(x) -> not fast(x)
- not bad(p)

Example

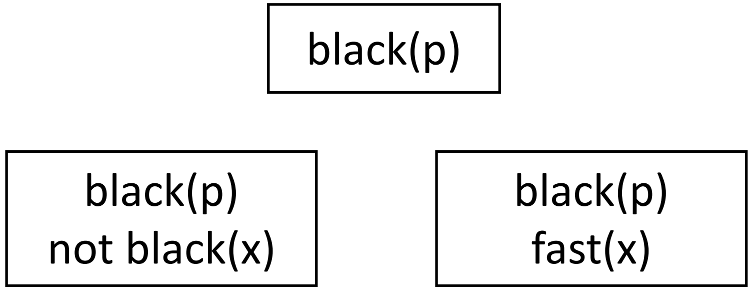
- black(p)
- for all x black(x) -> fast(x)
- for all x bad(x) -> not fast(x)
- bad(p)

Example

- black(p)
- not black(x) or fast(x)
- not bad(x) or not fast(x)
- bad(p)

Example

- black(p)
- not black(x) or fast(x)
- not bad(x) or not fast(x)
- bad(p)



UNIVERSITY OF CALGARY

Example

- black(p)
- not black(x) or fast(x)
- not bad(x) or not fast(x)
- bad(p)

mgu = {x = p}

black(p)

black(p)
not black(x)

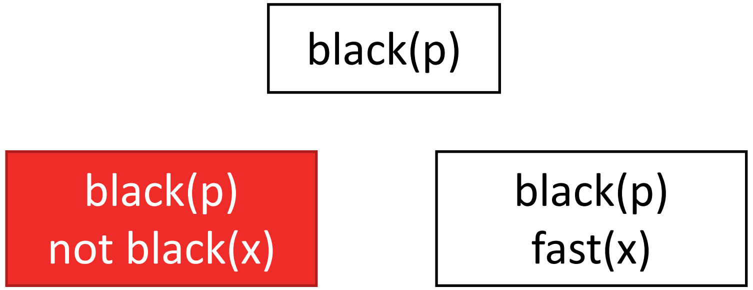
black(p)
fast(x)

UNIVERSITY OF CALGARY

Example

- black(p)
- not black(x) or fast(x)
- not bad(x) or not fast(x)
- bad(p)

mgu = {x = p}

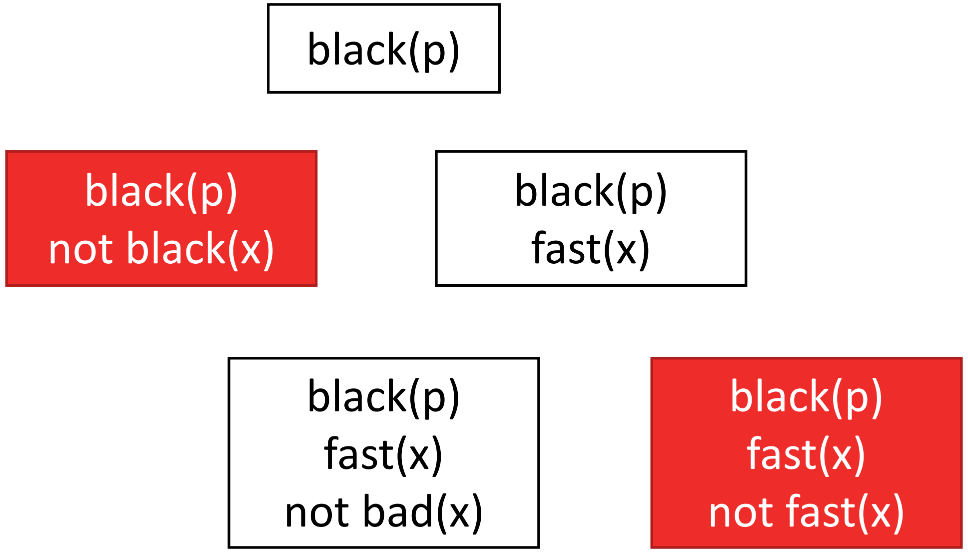


UNIVERSITY OF CALGARY

Example

- black(p)
- not black(x) or fast(x)
- not bad(x) or not fast(x)
- bad(p)

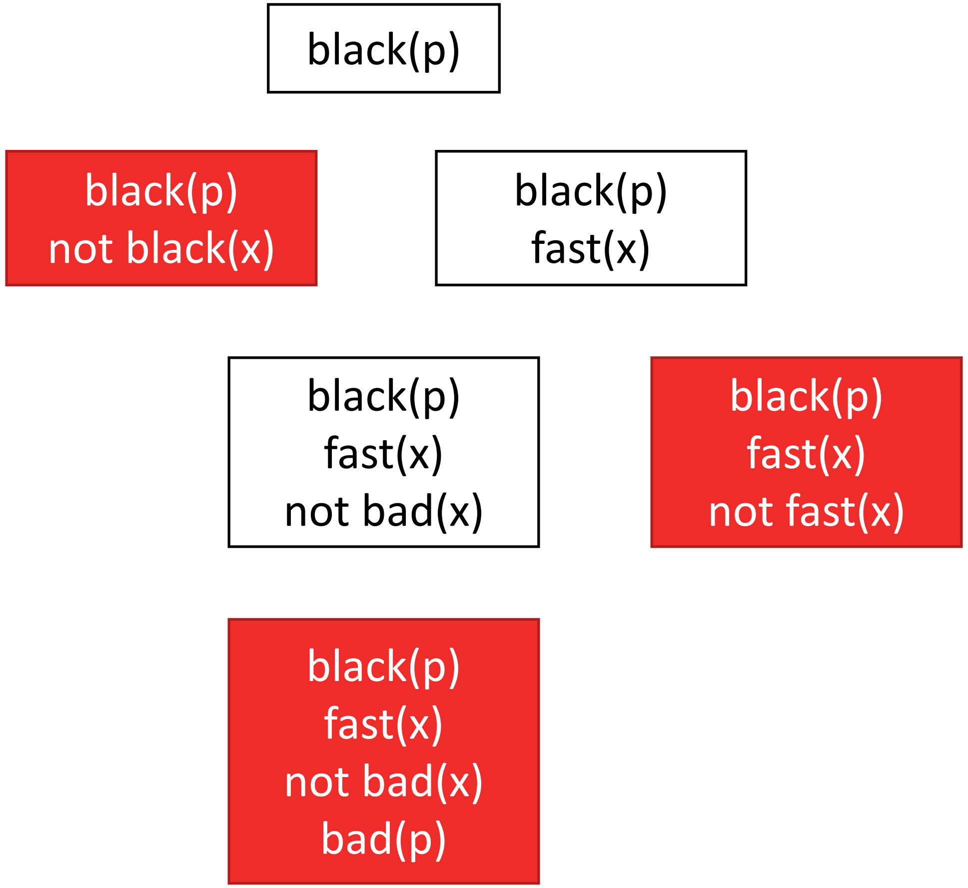
mgu = {x = p}



Example

- black(p)
- not black(x) or fast(x)
- not bad(x) or not fast(x)
- bad(p)

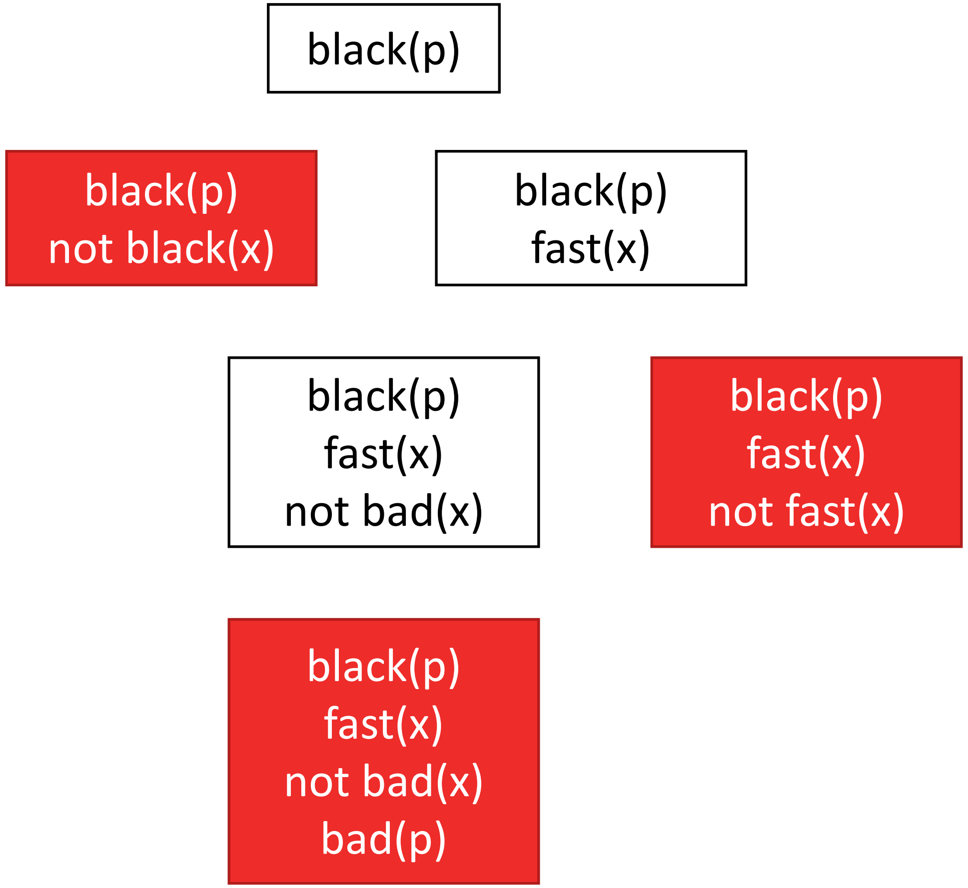
mgu = {x = p}



Example

- black(p)
- not black(x) or fast(x)
- not bad(x) or not fast(x)
- bad(p)

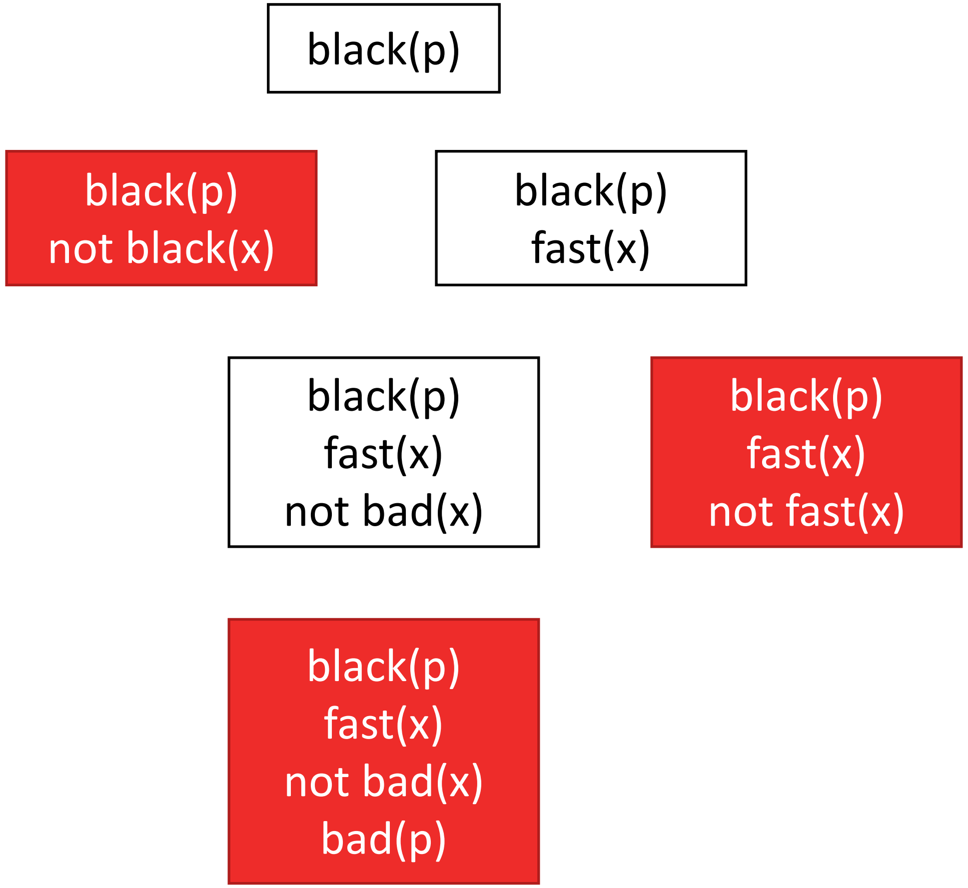
mgu = {x = p}



Example

- black(p)
- not black(x) or fast(x)
- not bad(x) or not fast(x)
- not bad(p)

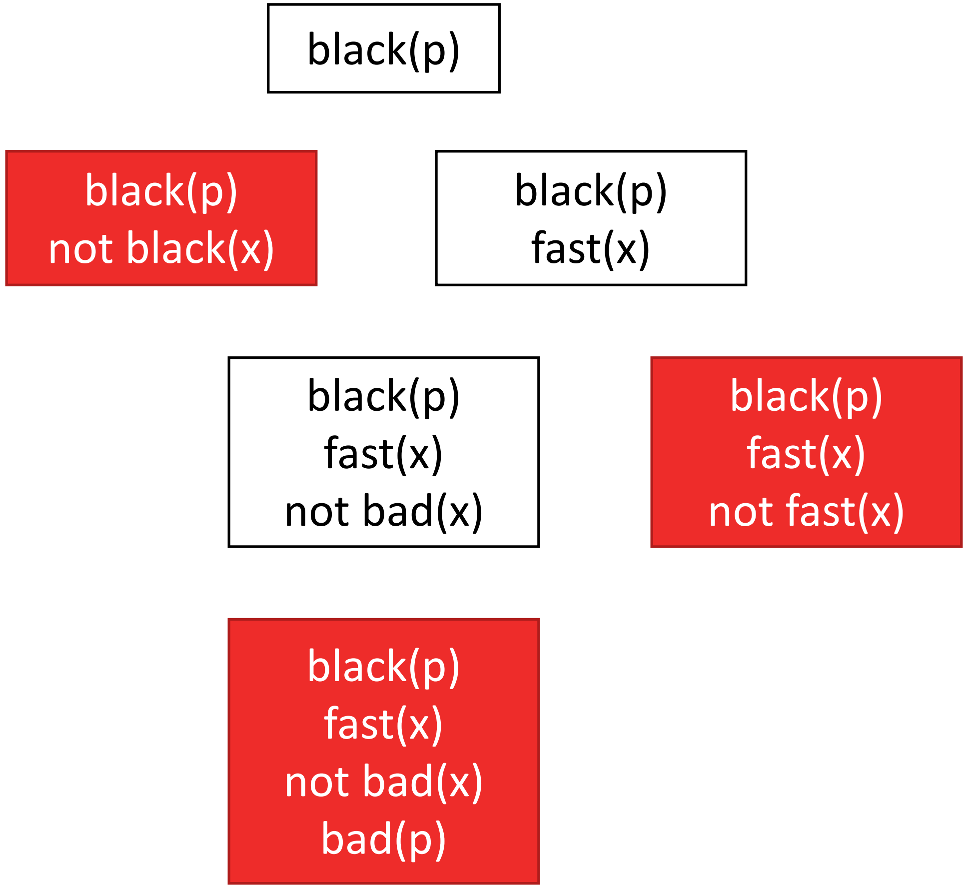
mgu = {x = p}



Example

- black(p)
- not black(x) or fast(x)
- not bad(x) or not fast(x)
- A Porsche is a good car.

mgu = {x = p}



Other Logics

Other logics

There are a lot of concepts that cannot be easily expressed in propositional logic or first-order predicate logic:

- Time
- Changes in the world
- Default values and overriding them
- Vagueness of information, fuzzy definitions and expressions, probabilities as truth values

"Modern" logics

- Modal logics: deal with time, changing worlds by having symbols based on a possible world structure (possible-world semantics)
- Nonmonotonic logics: allow for dealing with assumptions that later might be detected as false and then deals with the consequences of this by reevaluating everything that has been deduced so far (uses truth-maintenance systems)
- Multi-valued logics/fuzzy logics: allow for probabilistic reasoning, avoiding a black-and-white view of things

Onward to … rule systems

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

UNIVERSITY OF CALGARY