CS228 Logic for Computer Science 2021

Lecture 18: Terms and unification

Instructor: Ashutosh Gupta

IITB, India

Compile date: 2021-02-15

Topic 18.1

Game of terms

CNF formulas and proofs

Example 18.1

Recall we had a proof for $\emptyset \vdash (\forall x.\ (P(x) \lor Q(x)) \Rightarrow \exists x.P(x) \lor \forall x.Q(x))$.

Let us try to prove it via FOL CNF.

We fist take negation of the formula and transform it into FOL CNF. We obtain

$$\Sigma \triangleq \{\forall x.\ (P(x) \lor Q(x)), \forall x.\neg P(x), \neg Q(c)\}$$

We have written each clause as a separate formula without dropping quantifiers.

We show that we can derive contradiction from Σ.

CNF formulas and proofs

Recall

$$\Sigma \triangleq \{\forall x. \, (P(x) \lor Q(x)), \forall x. \neg P(x), \neg Q(c)\}$$

We can drive contradiction Here is a proof that derives contradiction.

$\forall x \neg P(x)$

1. $\Sigma \vdash \neg Q(c)$ — Assumption
2. $\Sigma \vdash \forall x. \, (P(x) \lor Q(x))$ — Assumption
3. $\Sigma \vdash P(x) \lor Q(x)$ — $\forall$-Elim applied to 2
4. $\Sigma \vdash \forall x. \, \neg P(x)$ — Assumption
5. $\Sigma \vdash \neg P(x)$ — $\forall$-Elim applied to 4
6. $\Sigma \vdash Q(x)$ — Resolution applied to 3 and 5
7. $\Sigma \vdash \forall x. \, Q(x)$ — $\forall$-Intro applied to 6
8. $\Sigma \vdash Q(c)$ — $\forall$-Elim applied to 7
9. $\Sigma \vdash Q(c) \land \neg Q(c)$ — $\land$-Intro applied to 1 and 8

Step 7 introduced c, which is a non-mechanical step, i.e., we need to plan to choose the term.

Example : an extreme example for finding a magic term.

Example 18.2

Let us derive contradiction from the following.
Let $\Sigma = \{\forall x_4, x_3, x_2, x_1.\ f(x_1, x_3, x_2) \neq f(g(x_2), j(x_4), h(x_3, a))\}$

Let us construct a proof for the above.

1. $\Sigma \vdash \forall x_4, x_3, x_2, x_1.\ f(x_1, x_3, x_2) \neq f(g(x_2), j(x_4), h(x_3, a))$
2. $\Sigma \vdash \forall x_3, x_2, x_1.\ f(x_1, x_3, x_2) \neq f(g(x_2), j(x_4), h(x_3, a))$ $\forall$*-Elim applied to 1*
3. $\Sigma \vdash \forall x_2, x_1.\ f(x_1, j(x_4), x_2) \neq f(g(x_2), j(x_4), h(j(x_4), a))$ $\forall$*-Elim applied to 2*
4. $\Sigma \vdash \forall x_1.\ f(x_1, j(x_4), h(j(x_4), a)) \neq f(g(h(j(x_4), a)), j(x_4), h(j(x_4), a))$ $\forall$*-Elim applied to 3*
5. $\Sigma \vdash f(g(h(j(x_4), a)), j(x_4), h(j(x_4), a)) \neq f(g(h(j(x_4), a)), j(x_4), h(j(x_4), a))$ $\forall$*-Elim applied to 4*

We need a mechanism to auto detect substitutions such that terms with variables become equal

Exercise 18.1

Finish the proof using Reflex and derive contradiction.

How to find the magic terms?

In the previous, example we were asked to equate terms

$$f(x_1, x_3, x_2) \text{ and } f(g(x_2), j(x_4), h(x_3, a))$$

by mapping variables x_1, x_2, x_3, and x_4 to terms.

The process of equating terms is called **unification**.

Sometimes, the unification may not even be possible.

Topic 18.2

Unification

Making terms equal by substitution

Unifier

Definition 18.1

For terms t and u, a substitution σ is a unifier of t and u if $t\sigma = u\sigma$.
We say t and u are unifiable if there is a unifier σ of t and u.

Example 18.3

Find a unifier σ of the following terms

- $x_4\sigma = f(x_1)\sigma$ $\quad\sigma = \{x_1 \mapsto c, x_4 \mapsto f(c)\}$
- $x_4\sigma = f(x_1)\sigma$ $\quad\sigma = \{x_1 \mapsto x_2, x_4 \mapsto f(x_2)\}$
- $g(x_1)\sigma = f(x_1)\sigma$ $\quad$ *not unifiable*
- $x_1\sigma = f(x_1)\sigma$ $\quad$ *not unifiable*

More general substitution

Commentary: The following definition depends on composition of substitution, which was discussed in earlier lectures. If not clear please look it up.

Definition 18.2

Let σ_1 and σ_2 be substitutions. σ_1 is more general than σ_2 if there is a substitution τ such that $\sigma_2 = \sigma_1\tau$. We write $\sigma_1 \geq \sigma_2$.

Example 18.4

- $\sigma_1 = \{x \mapsto f(y,z)\} \geq \sigma_2 = \{x \mapsto f(c, g(z))\}$ *because* $\sigma_2 = \sigma_1\{y \mapsto c, z \mapsto g(z)\}$.
- $\sigma_1 = \{x \mapsto f(y,z)\} \geq \sigma_2 = \{x \mapsto f(z,z)\}$ *because* $\sigma_2 = \sigma_1\{y \mapsto z\}$.

Exercise 18.2

If $\sigma_1 \geq \sigma_2$ and $\sigma_2 \geq \sigma_3$. Then, $\sigma_1 \geq \sigma_3$.

$\sigma_2 = \sigma_1 \tau_1$

$\sigma_3 = \sigma_2 \tau_2 = (\sigma_1 \tau_1) \tau_2 = \sigma_1 (\tau_1 \tau_2) = \sigma_1 \tau_3$

Most general unifier (mgu)

Is mgu unique? Does mgu always exist?

Definition 18.3

Let t and u be terms with variables, and σ be a unifier of t and u.
σ is most general unifier(mgu) of u and t if it is more general than any other unifier.

Example 18.5

Consider terms $f(x, g(y))$ and $f(g(z), u)$. The following are unifiers of the terms.

1. $\sigma_1 = \{x \mapsto g(z), u \mapsto g(y), z \mapsto z, y \mapsto y\}$
2. $\sigma_2 = \{x \mapsto g(c), u \mapsto g(d), z \mapsto c, y \mapsto d\}$
3. $\sigma_3 = \{x \mapsto g(z), u \mapsto g(z), z \mapsto z, y \mapsto z\}$

where c and d are constants.

Please note $\sigma_1 \geq \sigma_2$ and $\sigma_1 \geq \sigma_3$. $\sigma_2 \not\geq \sigma_3$ and $\sigma_3 \not\geq \sigma_2$.(why?)

Uniqueness of mgu

Definition 18.4

A substitution σ is a renaming if $\sigma : \text{Vars} \rightarrow \text{Vars}$ and σ is one-to-one

Theorem 18.1

$\sigma_1 \geq \sigma_2$

If σ_1 and σ_2 are mgus of u and t. Then there is a renaming τ such that $\sigma_1\tau = \sigma_2$.

Proof.

Since σ_1 is mgu, therefore there is a substitution $\hat{\sigma_1}$ such that $\sigma_2 = \sigma_1\hat{\sigma_1}$.
Since σ_2 is mgu, therefore there is a substitution $\hat{\sigma_2}$ such that $\sigma_1 = \sigma_2\hat{\sigma_2}$.
Therefore $\sigma_1 = \sigma_1\hat{\sigma_1}\hat{\sigma_2}$. (And also, $\sigma_2 = \sigma_2\hat{\sigma_2}\hat{\sigma_1}$.)

Without loss of generality, for each $y \in \text{Vars}$, if $y \notin FV(x\sigma_1)$ for each $x \in \text{Vars}$, then we assume $y\hat{\sigma_1} = y$.

Uniqueness of mgu (contd.)

Proof(contd.)

claim: for each $y \in \text{Vars}$, $y\hat{\sigma_1} \in \text{Vars}$

Consider a variable x such that $y \in FV(x\sigma_1)$. Three possibilities for $y\hat{\sigma_1}$.

1. If $y\hat{\sigma_1} = f(..)$, $x\sigma_1\hat{\sigma_1}$ is longer than $x\sigma_1$. Therefore, $x\sigma_1\hat{\sigma_1}\hat{\sigma_2}$ is longer than $x\sigma_1$.
   Contradiction.
2. If $y\hat{\sigma_1} = c$, $\hat{\sigma_2}$ will not be able to rename c back to y in $x\sigma_1$.
3. Therefore, we must have the third possibility, i.e., $y\hat{\sigma_1} \in \text{Vars}$ is a variable.

claim: for each $y_1 \neq y_2 \in \text{Vars}$, $y_1\hat{\sigma_1} \neq y_2\hat{\sigma_1}$

Assume $y_1\hat{\sigma_1} = y_2\hat{\sigma_1}$. $\hat{\sigma_2}$ will not be able to rename the variables back to distinct variables.(why?)
Contradiction.

$\hat{\sigma_1}$ is a renaming. □

Topic 18.3

Unification algorithm

How to find unifiers?

We need to identify where terms are not in agreement.

Apply substitutions to fix the disagreement.

Disagreement pairs

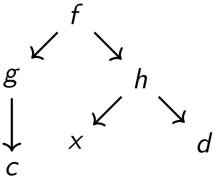
Definition 18.5

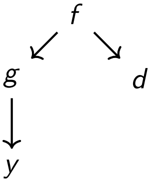
For terms t and u, d_1 and d_2 are disagreement pair if

1. *d_1 and d_2 are subterms of t and u respectively,*
2. *the path to d_1 in t is same as ~~and~~ the path to d_2 in u, and*
3. *roots of d_1 and d_2 are different.*

Example 18.6

Consider terms $t = f(g(c), h(x, d))$ and $u = f(g(y), d)$





Disagreement pairs: $h(x, d)$ and d

Disagreement pairs: c and y

Robinson algorithm for computing mgu

Algorithm 18.1: MGU($t, u \in T_S$)

$\sigma := \{\}$;
while $t\sigma \neq u\sigma$ **do**
  choose disagreement pair d_1, d_2 in $t\sigma$ and $u\sigma$;
  if *both* d_1 *and* d_2 *are non-variables* **then** **return** *FAIL* ;
  if $d_1 \in$ Vars **then**
    $x := d_1$; $s := d_2$;
  else
    $x := d_2$; $s := d_1$;
  if $x \in FV(s)$ **then** **return** *FAIL* ;
  $\sigma := \sigma\{x \mapsto s\}$ // update the substitution
return σ

If MGU is sound and always terminates then mgus for unifiable terms always exist.

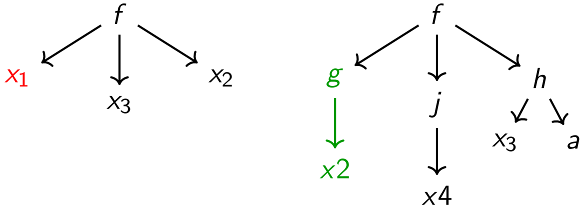
Exercise 18.3

Let $\sigma_0, \sigma_1, \ldots$ be the sequence of observed substitutions during the run of MGU*. Show $\sigma_i \geq \sigma_{i+1}$.*

Example: run of Robinson's algorithm

Example 18.7

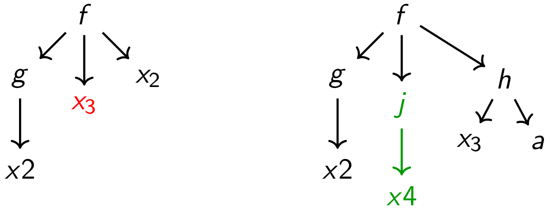
Consider call $\text{MGU}(f(x_1, x_3, x_2), f(g(x_2), j(x_4), h(x_3, a)))$
Initial $\sigma = \{\}$



Disagreement pairs $:= \{\ (x_1, g(x_2)),\ (x_3, j(x_4)),\ (x_2, h(x_3, a))\ \}$
Choose a disagreement pair: $(x_1, g(x_2))$
After update $\sigma = \{x_1 \mapsto g(x_2)\}$
Input terms after applying σ*:* $f(g(x_2), x_3, x_2)$ *and* $f(g(x_2), j(x_4), h(x_3, a))$

Example: run of Robinson's algorithm II (contd.)

Input terms now:



Disagreement pairs in the new terms:= $\{\ (x_3, j(x_4)),\ (x_2, h(x_3, a))\ \}$

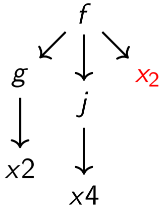
Choose a disagreement pair: $(x_3, j(x_4))$

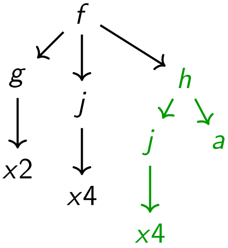
After update $\sigma = \{x_1 \mapsto g(x_2), x_3 \mapsto j(x_4)\}$

Input terms after applying σ: $f(g(x_2), j(x_4), x_2)$ and $f(g(x_2), j(x_4), h(j(x_4), a))$

Example: run of Robinson's algorithm III(contd.)

Input terms now:





Choose the last disagreement pair: $(x_2, h(j(x_4), a))$.

Since the mapping of x_1 refers to x_2 in old σ, it is also updated.

After applying new mapping $\sigma := \sigma\{x_2 \mapsto h(j(x_4), a)\}$

$$= \{x_1 \mapsto g(x_2), x_3 \mapsto j(x_4)\}\{x_2 \mapsto h(j(x_4), a)\}$$
$$= \{x_1 \mapsto g(h(j(x_4), a)), x_3 \mapsto j(x_4), x_2 \mapsto h(j(x_4), a)\}$$

Terms after applying σ: $f(g(h(j(x_4), a)), j(x_4), h(j(x_4), a))$ and $f(g(h(j(x_4), a)), j(x_4), h(j(x_4), a))$

Since no disagreement pairs, we are done.

Unification in proving

Example 18.8

Consider again $\forall x_1, x_2, x_3, x_4.\ f(x_1, x_3, x_2) \neq f(g(x_2), j(x_4), h(x_3, a))$

Given the above, one may ask

$$\textit{Are } f(x_1, x_3, x_2) \textit{ and } f(g(x_2), j(x_4), h(x_3, a)) \textit{ unifiable?}$$

If we run the unification algorithm on the above terms, we obtain

- $x_1 \mapsto g(h(j(x_4), a))$
- $x_2 \mapsto h(j(x_4), a)$
- $x_3 \mapsto j(x_4)$

We will integrate unification with a simpler resolution proof system.

The above instantiations are not magic anymore!

Topic 18.4

Correctness of Robinson algorithm

Termination of MGU

Theorem 18.2

MGU *always terminates.*

Proof.

Total number of variables in $t\sigma$ and $u\sigma$ decreases in every iteration.(why?)

Since initially there were finite variables in t and u, MGU terminates. $\square$

Soundness of MGU

Theorem 18.3

MGU(t, u) *returns unifier* σ *iff* t *and* u *are unifiable. Furthermore,* σ *is a mgu.*

Proof.

Since MGU must terminate, if t and u are not unifiable then MGU must return FAIL.

Let us suppose t and u are unifiable and τ is a unifier of t and u.
claim: $\tau = \sigma\tau$ is the loop invariant of MGU.

base case:
Initially, σ is identity. Therefore, the invariant holds initially.

induction step:
Induction hypothesis: $\tau = \sigma\tau$ holds at the loop head. ...

Soundness of MGU(contd.)

Proof(contd.)

claim: $t\sigma$ and $u\sigma$ are unifiable.

$$\underbrace{t\sigma\tau = t\tau}_{\text{Ind. Hyp.}} \underbrace{= u\tau}_{\text{Assumption}} \underbrace{= u\sigma\tau}_{\text{Ind. Hyp.}}.$$

claim: $x\tau = s\tau$.

Since $t\sigma\tau = u\sigma\tau$, and x and s are disagreement pairs in $t\sigma$ and $u\sigma$, $x\tau = s\tau$.

claim: $\{x \mapsto s\}\tau = \tau$.

Choose $y \in Vars$.

- If $y = x$, $y\{x \mapsto s\}\tau = s\tau = x\tau = y\tau$.
- If $y \neq x$, $y\{x \mapsto s\}\tau = y\tau$.

Therefore, $\{x \mapsto s\}\tau = \tau$.

Soundness of MGU(contd.)

Proof(contd.)

We now show that if we assume the invariant at the loop head, then FAIL is not returned.

claim: no FAIL at the first if condition
One of d_1 and d_2 is a variable. Otherwise $t\sigma$ and $u\sigma$ are not unifiable.

claim: no FAIL at the last if condition
Since $x\tau = s\tau$, x cannot occurs in s. Otherwise, no unifier can make them equal(why?).

...

Soundness of MGU(contd.)

Proof(contd.)

Since there is no fail, we show that invariant will continue to hold after the iteration.

claim: $\sigma\{x \mapsto s\}\tau = \tau$

Since $\{x \mapsto s\}\tau = \tau$, $\sigma\{x \mapsto s\}\tau = \sigma\tau$. By induction hypothesis, $\sigma\{x \mapsto s\}\tau = \tau$.

Due to the invariant $\tau = \sigma\tau$, σ is mgu at the termination. $\square$

Topic 18.5

Problems

MGU

Exercise 18.4

Find mgu of the following terms

1. $f(g(x_1), h(x_2), x_4)$ *and* $f(g(k(x_2, x_3)), x_3, h(x_1))$
2. $f(x, y, z)$ *and* $f(y, z, x)$
3. $\textsc{MGU}(f(g(x), x), f(y, g(y)))$

Exercise 18.5

Let σ_1 and σ_2 be the MGUs in the above exercise. Give unifiers σ_1' and σ_2' for the problems respectively such that they are not MGUs. Also give τ_1 and τ_2 such that

1. $\sigma_1' = \sigma_1\tau_1$
2. $\sigma_2' = \sigma_2\tau_2$

Maximum and minimal substitutions

Exercise 18.6

a. *Give two maximum general* substitutions *and two minimal general* substitutions*.*

b. *Show that maximum general* substitutions *are equivalent under renaming.*

Multiple unification

Definition 18.6

Let $t_1, .., t_n$ be terms. A substitution σ is a unifier of $t_1, .., t_n$ if $t_1\sigma = .. = t_n\sigma$.
We say $t_1, .., t_n$ are unifiable if there is a unifier σ of them.

Exercise 18.7

Write an algorithm for computing multiple unifiers using the binary MGU.

Concurrent unification

Definition 18.7

Let $t_1, .., t_n$ and $u_1, .., u_n$ be terms. A substitution σ is a concurrent unifier of $t_1, .., t_n$ and $u_1, .., u_n$ if

$$t_1\sigma = u_1\sigma, \quad .., \quad t_n\sigma = u_n\sigma.$$

We say $t_1, .., t_n$ and $u_1, .., u_n$ are concurrently unifiable if there is a unifier σ for them.

Exercise 18.8

Write an algorithm for concurrent unifiers using the binary $\textsc{MGU}$.

Topic 18.6

Extra slides: algorithms for unification

Robinson is exponential

Robinson algorithm has worst case exponential run time.

Example 18.9

Consider unification of the following terms

$f(x_1, g(x_1, x_1), x_2, ...)$

$f(g(y_1, y_1), y_2, g(y_2, y_2), ...)$

The mgu:

- $x_1 \mapsto g(y_1, y_1)$
- $y_2 \mapsto g(g(y_1, y_1), g(y_1, y_1))$
- .... *(size of term keeps doubling)*

After discovery of a substitution $x \mapsto s$, Robinson checks if $x \in FV(s)$. Therefore, Robinson has worst case exponential time.

Martelli-Montanari algorithm

This algorithm is lazy in terms of applying occurs check

Algorithm 18.2: MM-MGU($t, u \in T_S$)

$\sigma := \lambda x.x$; $M = \{t = u\}$;
while *change in M or σ* **do**
  if $f(t_1, ...t_n) = f(u_1, ...u_n) \in M$ **then**
    $M := M \cup \{t_1 = u_1, ...t_n = u_n\} - \{f(t_1, ...t_n) = f(u_1, ...u_n)\}$;
  if $f(t_1, ...t_n) = g(u_1, ...u_n) \in M$ **then** **return** *FAIL* ;
  if $x = x \in M$ **then** $M := M - \{x = x\}$;
  if $x = t' \in M$ *or* $t' = x \in M$ **then**
    if $x \in FV(t')$ **then** **return** *FAIL* ;
    $\sigma := \sigma[x \mapsto t']$; $M := M\sigma$
return σ

Commentary: Please find more details on https://pdfs.semanticscholar.org/3cc3/338b59855659ca77fb5392e2864239c0aa75.pdf

Escalada-Ghallab Algorithm

There is also Escalada-Ghallab Algorithm for unification.

End of Lecture 18