CS228 Logic for Computer Science 2020

Lecture 18: Terms and unification

Instructor: Ashutosh Gupta

IITB, India

Compile date: 2020-02-21

Topic 18.1

Game of terms

Terms need matching

The evidence of $\exists$ is usually a term, which need to be identified.
We need to start proving accordingly.

Here is a simple example.

Example 18.1

Let us prove $\emptyset \vdash H(f(c)) \Rightarrow \exists x.\ H(x)$

1. $\{H(f(c))\} \vdash H(f(c))$ *Assumption*
2. $\emptyset \vdash H(f(c)) \Rightarrow \exists x.H(x)$ $\exists$*-Intro applied to 1*
3. $\emptyset \vdash H(f(c)) \Rightarrow \exists x.H(x)$ $\Rightarrow$*-Intro applied to 2*

Example : finding evidence of $\exists$ is hard

There are magic terms that can provide evidence of $\exists$. Here is an extreme example.

Example 18.2

Consider $\emptyset \vdash \exists x_1, x_2, x_3, x_4.\ f(x_1, x_3, x_2) = f(g(x_2), j(x_4), h(x_3, a))$
Let us construct a proof for the above as follows

1. $\emptyset \vdash f(g(h(j(c), a)), j(c), h(j(c), a)) = f(g(h(j(c), a)), j(c), h(j(c), a))$
2. $\emptyset \vdash \exists x_4. f(g(h(j(x_4), a)), j(x_4), h(j(x_4), a)) = f(g(h(j(x_4), a)), j(x_4), h(j(x_4), a))$
3. $\emptyset \vdash \exists x_3, x_4.\ f(g(h(x_3, a)), x_3, h(x_3, a)) = f(g(h(x_3, a)), j(x_4), h(x_3, a))$
4. $\emptyset \vdash \exists x_2, x_3, x_4.\ f(g(x_2), x_3, x_2) = f(g(x_2), j(x_4), h(x_3, a))$
5. $\emptyset \vdash \exists x_1, x_2, x_3, x_4.\ f(x_1, x_3, x_2) = f(g(x_2), j(x_4), h(x_3, a))$

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

How to find the magic terms?

In the previous example we were needed 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

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

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

More general substitution

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 $\sigma_2 = \sigma_1\tau$.

Example 18.4

- $\sigma_1 = \{x \mapsto f(y,z)\}$ *is more general than* $\sigma_2 = \{x \mapsto f(c, g(z))\}$ *because* $\sigma_2 = \sigma_1\{y \mapsto c, z \mapsto g(z)\}$
- $\sigma_1 = \{x \mapsto f(y,z)\}$ *is more general than* $\sigma_2 = \{x \mapsto f(z,z)\}$ *because* $\sigma_2 = \sigma_1\{y \mapsto z\}$

Exercise 18.1

If σ_1 is more general than σ_2 and σ_2 is more general than σ_3. Then, σ_1 is more general than σ_3.

Most general unifier (mgu)

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 $f(g(z), u)$
Consider the following three unifiers

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

The first is more general unifier than the bottom two.
The second and third are incomparable(why?)

Is mgu unique? Does mgu always exist?

Uniqueness of mgu

Definition 18.4

*A substitution σ is a renaming if σ : **Vars** $\rightarrow$ **Vars** and σ is one-to-one*

Theorem 18.1

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 $\sigma_1 = \sigma_1\hat{\sigma_1}\hat{\sigma_2}$.

Wlog, for each $y \in$ **Vars**, if $y \notin FV(x\sigma_1)$ for each $x \in$ **Vars**, then we assume $y\hat{\sigma_1} = y$

Uniqueness of mgu (contd.)

Proof(contd.)

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

Consider a variable x s.t. $y \in FV(x\sigma_1)$.

Three possibilities for $y\hat{\sigma_1}$.

Suppose $y\hat{\sigma_1} = f(..)$: $x\sigma_1\hat{\sigma_1}$ will be longer than $x\sigma_1$.

Therefore, $x\sigma_1\hat{\sigma_1}\hat{\sigma_2}$ will be longer than $x\sigma_1$. Contradiction.

Suppose $y\hat{\sigma_1} = c$: $\hat{\sigma_2}$ will not be able to rename c back to y in $x\sigma_1$.

Therefore $y\hat{\sigma_1} \in \textbf{Vars}$ is variable.

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

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?) $\square$

Attendance quiz

Which of the following are true about substitutions?

$\sigma = \{x \mapsto y, y \mapsto x\}$ is renaming
$\sigma = \{x \mapsto x, y \mapsto y\}$ is renaming
$\sigma = \{x \mapsto f(y)\}$ unifies $g(x)$ and $g(f(y))$
x and $f(x)$ are not unifiable
$g(x)$ and $f(x)$ are not unifiable
$\{x \mapsto x, y \mapsto y\}$ is more general than $\{x \mapsto y, y \mapsto y\}$.
$\{x \mapsto x, y \mapsto y\}$ is more general than $\{x \mapsto f(x), y \mapsto y\}$.
$\sigma = \{x \mapsto f(y), y \mapsto y\}$ is renaming
$\sigma = \{x \mapsto y, y \mapsto y\}$ is renaming
$\sigma = \{x \mapsto f(y)\}$ unifies $g(x)$ and $g(f(x))$
x and $f(x)$ are unifiable
$g(x)$ and $f(x)$ are unifiable
$\{x \mapsto y, y \mapsto y\}$ is more general than $\{x \mapsto x, y \mapsto y\}$.
$\{x \mapsto f(x), y \mapsto y\}$ is more general than $\{x \mapsto x, y \mapsto y\}$.

Disagreement pair

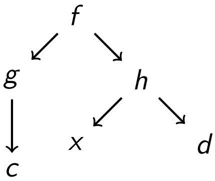
Definition 18.5

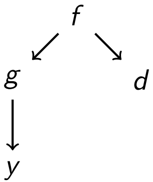
For terms t and u, d_1 and d_2 are disagreement pair if

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 $u = f(g(y), d)$
(Node labels are pairs of function symbols and argument number)





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

Disagreement pairs: c and y

Robinsion's algorithm for computing mgu

Algorithm 18.1: MGU($t, u \in T_{\mathbf{S}}$)

$\sigma := \{\}$;
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* ;
  $\sigma := \sigma\{x \mapsto s\}$ // remove x in terms of σ and assign s to x
return σ

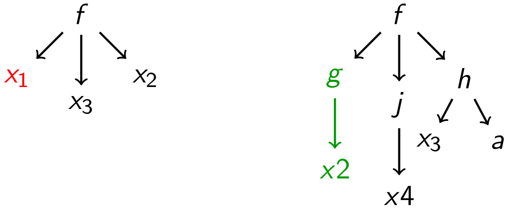
If MGU is sound and always terminates then mgus for unifiable terms always exist.

Example: run of Robinsion's algorithm

Example 18.7

Consider call $\mathrm{MGU}(f(x_1, x_3, x_2), f(g(x_2), j(x_4), h(x_3, a)))$

Initial $\sigma = \{\}$

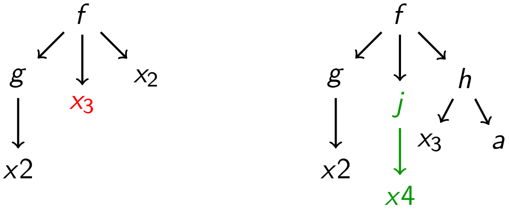


Disagreement pairs $:= \{\ (x_1, g(x_2)),\ (x_3, j(x_4)),\ (x_2, h(x_3, a))\ \}$

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 Robinsion's algorithm II (contd.)

Input terms now:

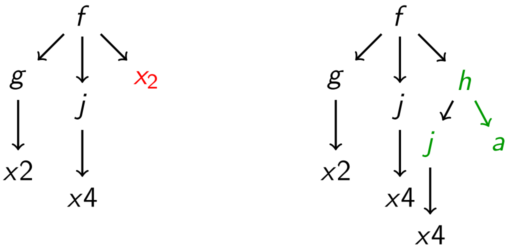


Disagreement pairs the new terms:$=\{\ (x_3, j(x_4)),\ (x_2, h(x_3, a))\ \}$
Choose a disagreement pair: $(x_3, j(x_4))$
After update $\sigma = \{x_1 \mapsto g(x_2), x_3 \mapsto j(x_4)\}$
Input terms after applying σ:

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

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

Input terms now:



Choose the only remaining disagreement pair: $(x_2, h(j(x_4), a))$.
Current $\sigma = \{x_1 \mapsto g(x_2), x_3 \mapsto j(x_4)\}$ refers to x_2.
After applying new mapping
$\sigma := \sigma\{x_2 \mapsto h(j(x_4), a)\} = \{x_1 \mapsto g(h(j(x_4), a)), x_3 \mapsto j(x_4)\}$
After including the new mapping
$\sigma = \{x_1 \mapsto g(h(j(x_4), a)), x_3 \mapsto j(x_4), x_2 \mapsto h(j(x_4), a)\}$
Terms after applying σ:

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

No disagreement pairs in the new terms. Therefore, we are done.

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:
We assume $\tau = \sigma\tau$ holds at the loop head. ...

Soundness of MGU(contd.)

Proof(contd.)

We show that the invariant holds after the loop body and FAIL is not returned.

claim: no FAIL at the first **if**

$t\sigma$ and $u\sigma$ are unifiable because $t\sigma\tau = t\tau = u\tau = u\sigma\tau_{\text{(why?)}}$.

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**

Since $t\sigma\tau = u\sigma\tau$, $x\tau = s\tau$.

If x occurs in s then no unifier can make them equal(why?).

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

$x\sigma\{x \mapsto s\}\tau = \underbrace{s\tau = x\tau}_{\text{Proven in the last claim}}$.

Let $y \neq x$ then $\underbrace{y\sigma\{x \mapsto s\}\tau = y\sigma\tau}_{(why?)} = y\tau$.

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

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

Multiple unification

Definition 18.6

Let $t_1, .., t_n$ be terms with variables.
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 18.2

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 with variables.
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 18.3

Write an algorithm for concurrent unifiers using the binary MGU.

Topic 18.3

Unification in proving

Unification in proving

Example 18.8

Consider again

$\emptyset \vdash \exists x_1, x_2, x_3, x_4.\ f(x_1, x_3, x_2) = f(g(x_2), j(x_4), h(x_3, a))$

Given the above, one may ask

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

Exercise 18.4

Run the unification algorithm on the above terms

Answer:

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

We will integrate unification to a far simpler resolution proof system.

The above instantiations are not magic anymore!

Topic 18.4

Problems

MGU

Exercise 18.5

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 18.6

Let σ_1 and σ_2 be the MGUs in the above unifications. 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 18.7

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

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

Topic 18.5

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: $\textsc{MM-Mgu}(t, u \in T_{\mathbf{S}})$

$\sigma := \lambda x.x$; $M = \{t = u\}$;
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

https://pdfs.semanticscholar.org/3cc3/338b59855659ca77fb5392e2864239c0aa75.pdf

Escalada-Ghallab Algorithm

There is also Escalada-Ghallab Algorithm for unification.

End of Lecture 18