

INF4170 – Logikk

Forelesning 9: Automatisk bevissøk –
introduksjon, substitusjoner og unifisering

Martin Giese

Institutt for informatikk, Universitetet i Oslo

7. april 2008



Dagens plan

1 Automatisk bevissøk

Automatisk bevissøk i førsteordens logikk

- Sekventkalkylen LK tilbyr
 - et sett med regler for å bygge opp utledninger, og
 - en egenskap som skiller bevis fra utledninger.
- **Sunnhet** sikrer oss at enhver bevisbar sekvent er gyldig.
- **Kompletthet** sikrer oss at det *finnes* et bevis for enhver gyldig sekvent.
- Kalkylen sier imidlertid ingenting om **hvordan** man finner bevis for gyldige sekventer!
- Kompletthetsbeviset for LK gir hint om hvordan vi kan lage en søkealgoritme.
- La oss forsøke!

Noen begreper

- En utledning er **lukket** hvis alle grenene er lukket.
- En utledning er **utvidbar** hvis det er mulig å anvende en regel på en formel i en løvsekvent i utledningen.
- En søkealgoritme er **komplett** hvis den *finner* et bevis for enhver gyldig sekvent.

Algoritme: gyldig? ($\Gamma \vdash \Delta$)

```

 $\pi := \Gamma \vdash \Delta$ ;
while ( $\pi$  ikke er lukket) do
  if ( $\pi$  ikke er utvidbar) then
    return "ikke gyldig";
  else
     $\varphi :=$  ikke-atomær formel i løvsekvent i  $\pi$ ;
    utvid  $\pi$  ved å anvende riktig LK-regel på  $\varphi$ ;
  end if
end while
return "gyldig";

```

- Algoritmen er komplett hvis utvelgelsen av φ er **rettferdig**.

Effektivitet

- Effektiviteten til algoritmen avhenger av tre ting:
 - Hvor effektivt er det å sjekke om utledningen er lukket?
 - Strategi for valg av utvidelse av utledningen.
 - Hvor effektiv er selve utvidelsen, dvs. regelanvendelsen?
- I første runde ser vi på punkt 1 og 3.
- Senere introduseres **koblingskalkylen**, som gir oppgav til en strategi for valg av utvidelser av utledningene.
- La oss starte med punkt 3 – effektiviteten til regelanvendelsene.

Hvor kostbare er regelanvendelsene?

- α - og β -reglene henter ut delformler fra en sammensatt formel:

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} L\wedge \quad \frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta} R\wedge$$

- All nødvendig informasjon tilgjengelig i hovedformelen: kan utføres i konstant tid.
- Riktignok får vi en del formelkopiering i β -regelen, men dette kan optimaliseres med f.eks. pekere i en objektorientert implementasjon.
- δ -regelen setter inn en ny parameter for den bundne variabelen:

$$\frac{\Gamma, \varphi[a/x] \vdash \Delta}{\Gamma, \exists x \varphi \vdash \Delta} L\exists$$

- Parametrene kan nummereres: utføres i lineær tid.

γ -reglene

- La oss se på γ -reglene:

$$\frac{\Gamma, \forall x \varphi, \varphi[t/x] \vdash \Delta}{\Gamma, \forall x \varphi \vdash \Delta} L\forall \quad \frac{\Gamma \vdash \Delta, \exists x \varphi, \varphi[t/x]}{\Gamma \vdash \Delta, \exists x \varphi} R\exists$$

- Vi kan sette inn en vilkårlig lukket term t for x .
- For å få en komplett algoritme, må vi (før eller senere) instansiere hver γ -formel med **alle** termene i Herbranduniverset.
- Vi kan nummerere termene i Herbranduniverset og instansiere γ -formlene i denne rekkefølgen.
- Hvilken rekkefølge er gunstig med tanke på å **finne bevis så tidlig som mulig**?

$$\forall x P x, P a, \dots, P f f a \vdash P f f a, Q g a$$

$$\frac{\vdots}{\forall x P x, P a \vdash P f f a, Q g a} \quad a, f a, g a, f f a, f g a, \dots, f f f a, \dots$$

1 2 3 4 5 i

Utsette valg av γ -term

- En bedre idé: Utsette valg av term i γ -reglene til et senere tidspunkt.
- La γ -reglene sette inn **frie variable**:

$$\frac{\frac{a/u}{\forall xPx, Pu \vdash Pa} \quad \frac{b/v}{\forall xPx, Pv \vdash Pb}}{\forall xPx \vdash Pa \wedge Pb}$$

- Substituere termer for variable slik at løvnodene blir aksiomer.
- Hvilke substitusjoner vi kan anvende på løvnoder med frie variable slik at de blir aksiomer?
- Problemet kan løses med **unifiseringsalgoritmer**.

 δ -reglene

- Når vi setter inn variable i γ -reglene får vi imidlertid problemer med δ -reglene.
- Hvordan sikre at parameteren vi setter inn er **ny** når vi ennå ikke har satt inn termer for de frie variablene?

$$\frac{\frac{b/u, a/v}{Lu \vdash Lbv}}{\frac{\exists yLuy \vdash \forall xLxv}{\forall x\exists yLxy \vdash \exists y\forall xLxy}}$$

kan ikke lukkes

$$\frac{Luf(u) \vdash Lg(v)v}{\exists yLuy \vdash \forall xLxv}$$

$$\frac{\exists yLuy \vdash \forall xLxv}{\forall x\exists yLxy \vdash \exists y\forall xLxy}$$

- Vi lar δ -reglene introdusere en **Skolemterm**:

$$f(u_1, \dots, u_n),$$

der f er et nytt funksjonssymbol, kalt en **Skolemfunksjon**, og u_1, \dots, u_n er alle variablene som forekommer fritt i δ -formelen.

- På den måten sikrer vi at termen introdusert av δ -regelen er **ny** uansett hva slags verdi vi velger å instansiere de frie variablene med.

Oppsummering

- Vi skal introdusere en **fri-variabel sekventkalkyle** og vise at den er **sunn** og **komplett**.
- γ -reglene introduserer nye **frie variable** og δ -reglene introduserer **Skolemtermer**.
- Ved hjelp av **unifiseringsalgoritmer** finner vi **substitusjoner** som **lukker** utledningen.

1 Automatisk bevisøk

- Introduksjon
- Substitusjoner
- Unifisering

Substitusjoner

- Vi har tidligere definert $\varphi[s/x]$ som formelen vi får ved å erstatte alle frie forekomster av x i φ med s .
- I fri-variabel sekventkalkyle har vi behov for å erstatte flere forskjellige variable med termer **samtidig**.
- Vi skal nå definere en bestemt type funksjoner – **substitusjoner** – som generaliserer én-variabel substitusjon til flere variable.
- Notasjon: Når vi anvender en substitusjon σ på en formel φ eller en term t skriver vi $\varphi\sigma$ eller $t\sigma$ istedenfor $\sigma(\varphi)/\sigma(t)$.

Definisjon (Substitusjon)

En **substitusjon** er en funksjon σ fra mengden variable \mathcal{V} til mengden av termer \mathcal{T} i et gitt førsteordens språk.

- **Støtten** (support) eller **støttemengden** (support set) til σ er mengden av variable x slik at $x\sigma \neq x$.
- σ er **grunn** dersom $x\sigma$ er en lukket term for alle variable x i støttemengden til σ .

Notasjon

En substitusjon σ med endelig støtte $\{x_1, \dots, x_n\}$ slik at $x_1\sigma = t_1, \dots, x_n\sigma = t_n$ skriver vi ofte slik:

$$\sigma = \{t_1/x_1, \dots, t_n/x_n\}$$

- Substitusjonen ϵ slik at $x\epsilon = x$ for alle variable x kalles **identitetssubstitusjonen**.
- Identitetssubstitusjonen kan skrives $\{\}$ siden den har tom støttemengde.

$$\sigma = \{a/x, fa/y\}$$

- er en substitusjon slik at
 - $x\sigma = a$
 - $y\sigma = fa$
 - $z\sigma = z$ for alle andre variable
- er en grunn substitusjon

$$\tau = \{a/y, fx/z\}$$

- er en substitusjon slik at
 - $y\tau = a$
 - $z\tau = fx$
 - $v\tau = v$ for alle andre variable
- er **ikke** en grunn substitusjon

Substitusjon på termer

- Vi definerer substitusjon på termer som tidligere.

Definisjon (Substitusjon på termer)

Vi definerer resultatet av å anvende en substitusjon σ på vilkårlige termer rekursivt ved:

- $c\sigma = c$ for et konstantsymbol c .
- $f(t_1, \dots, t_n)\sigma = f(t_1\sigma, \dots, t_n\sigma)$ for en funksjonsterm $f(t_1, \dots, t_n)$.

La $\sigma = \{gy/x, y/z\}$.

- $f(x, a)\sigma = f(gy, a)$
- $h(y, z)\sigma = h(y, y)$
- $x\sigma = gy$

La $\tau = \{y/x, x/y\}$.

- $x\tau = y$
- $f(x, y)\tau = f(y, x)$

Substitusjon på formler

- Som tidligere, ønsker vi at substitusjoner **ikke** skal endre **bundne** variable.
- Eksempel: for $\sigma = \{a/x, b/y\}$ så vil $\forall x(Px \rightarrow Qy)\sigma = \forall xPx \rightarrow Qb$.
- Vi begrenser substitusjonen på den bundne variabelen:

Definisjon (Begrenset substitusjon)

La σ være en substitusjon. Substitusjonen σ **begrenset** på x , skrevet σ_x , er definert slik at

$$y\sigma_x = \begin{cases} y & \text{hvis } y = x \\ y\sigma & \text{ellers} \end{cases}$$

for enhver variabel y .

Definisjon (Substitusjon på formler)

$\varphi\sigma$ er definert rekursivt ved:

- 1 $R(t_1, \dots, t_n)\sigma = R(t_1\sigma, \dots, t_n\sigma)$
- 2 $\neg\psi\sigma = \neg(\psi\sigma)$
- 3 $(\varphi_1 \circ \varphi_2)\sigma = (\varphi_1\sigma \circ \varphi_2\sigma)$, hvor $\circ \in \{\wedge, \vee, \rightarrow\}$
- 4 $(Qx\psi)\sigma = Qx(\psi\sigma_x)$, hvor $Q \in \{\forall, \exists\}$

- Vi antar, som tidligere, at ingen variable blir bundet som resultat av å anvende en substitusjon.
- Dette kan vi unngå ved å omdøpe bundne variable.

La $\sigma = \{fx/x, a/y, y/z\}$

- $\sigma_x = \{\cancel{fx/x}, a/y, y/z\}$
- $\sigma_y = \{fx/x, \cancel{a/y}, y/z\}$
- $\sigma_z = \{fx/x, a/y, \cancel{y/z}\}$
- $P(x, y)\sigma = P(fx, a)$
- $\forall xP(x, y)\sigma = \forall x(P(x, y)\sigma_x) = \forall xP(x, a)$
- $\exists z(Px \rightarrow Qz)\sigma = \exists z((Px \rightarrow Qz)\sigma_z) = \exists z(Pfx \rightarrow Qz)$

Komposisjon av substitusjoner

- La σ og τ være substitusjoner.
- Anta at vi først anvender σ og så τ på en formel φ : $(\varphi\sigma)\tau$.
- Vi har av og til bruk for å snakke om den substitusjonen som tilsvarer å anvende σ etterfulgt av τ .

Definisjon (Komposisjon av substitusjoner)

La σ og τ være substitusjoner. **Komposisjonen** av σ og τ er en substitusjon skrevet $\sigma\tau$ slik at $x(\sigma\tau) = (x\sigma)\tau$ for hver variabel x .

- Oppgave: vis at $\varphi(\sigma\tau) = (\varphi\sigma)\tau$ for alle formel φ og alle substitusjoner σ og τ .

Komposisjon av substitusjoner med endelig støtte

Påstand

La $\sigma_1 = \{s_1/x_1, \dots, s_n/x_n\}$ og $\sigma_2 = \{t_1/y_1, \dots, t_k/y_k\}$. Da er

$$\sigma_1\sigma_2 = \{(s_1\sigma_2)/x_1, \dots, (s_n\sigma_2)/x_n, (z_1\sigma_2)/z_1, \dots, (z_m\sigma_2)/z_m\}$$

der z_1, \dots, z_m er de variablene blant y_1, \dots, y_k som **ikke** er blant x_1, \dots, x_n .

La $\sigma = \{z/x, a/y\}$ og $\tau = \{b/y, a/z\}$.

Da er $\sigma\tau = \{(z\tau)/x, (a\tau)/y, (z\tau)/z\} = \{a/x, a/y, a/z\}$.

La $\sigma = \{y/x\}$ og $\tau = \{x/y\}$.

Da er $\sigma\tau = \{(y\tau)/x, (y\tau)/y\} = \{x/x, x/y\} = \{x/y\}$.

1 Automatisk bevisssøk

- Introduksjon
- Substitusjoner
- Unifisering

Unifisering

- I fri-variabel sekventkalkyle kan vi ha løvsekventer på formen

$$\Gamma, P(s_1, \dots, s_n) \vdash P(t_1, \dots, t_n), \Delta$$

der hver s_i og t_i er termer som kan inneholde variable.

- For å lukke løvsekventen må vi finne en substitusjon σ slik at $s_i\sigma = t_i\sigma$ for hver i .
- **Det er ikke sikkert at noen slik substitusjon finnes!**

Unifiseringsproblemet

La s og t være termer. Finn *alle* substitusjoner som gjør s og t syntaktisk like, dvs. alle σ slik at $s\sigma = t\sigma$.

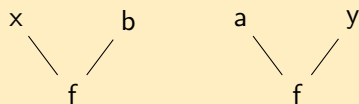
- En substitusjon som gjør termene s og t syntaktisk like, kalles en **unifikator** for s og t .
- To termer er **unifiserbare** hvis de har en unifikator.

Er $f(x)$ og $f(a)$ unifiserbare?

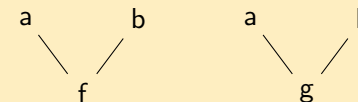
Ja. Vi ser at $\sigma = \{a/x\}$ er en *unifikator*: $f(x)\sigma = f(a)$

Er $f(x, b)$ og $f(a, y)$ unifiserbare?

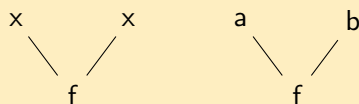
Kan v re lettere   se hvis vi skriver termene som *tr er*:



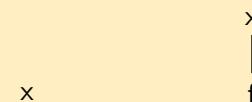
- Symbolene i posisjon 0 (rotposisjonen) er like.
- Symbolene i venstre barn er ulike, men kan unifiseres med $\{a/x\}$.
- Symbolene i h yre barn er ulike, men kan unifiseres med $\{b/y\}$.

Er $f(a, b)$ og $g(a, b)$ unifiserbare?

- Symbolene i posisjon 0 er ulike, og kan *ikke* unifiseres!

Er $f(x, x)$ og $f(a, b)$ unifiserbare?

- Symbolene i posisjon 0 er like.
- Symbolene i venstre barn er ulike, men kan unifiseres med $\{a/x\}$.
- Vi m  anvende $\{a/x\}$ p  x i b de venstre og h yre barn.
- Symbolene i h yre barn er n  ulike, og kan *ikke* unifiseres!

Er x og $f(x)$ unifiserbare?

- Symbolene i posisjon 0 er ulike, men kan unifiseres med $\{f(x)/x\}$.
- Vi m  samtidig anvende $\{f(x)/x\}$ p  x i h yre tre.
- P  posisjon 1 ser vi n  at symbolene x og f er ulike.
- Hvis vi unifiserer med $\{f(x)/x\}$, m  vi igjen erstatte x i h yre tre.
- S nn kan vi holde p  en stund...

Generelt har vi:

- To *ulike* konstantsymboler eller funksjonssymboler er **ikke** unifiserbare.
 - En variabel x er **ikke** unifiserbar med en term som *inneholder* x .
-
- Vi skal lage en **unifiseringsalgoritme**, som finner **alle** unifikatorer for to termer.
 - Problem: To termer har potensielt uendelig mange unifikatorer! Vi kan ikke returnere alle...
 - Løsning: Finne en **representant** σ for mengden av unifikatorer slik at alle andre unifikatorer kan konstrueres fra σ .
 - En slik unifikator kalles en **mest generell unifikator**.

Definisjon (Mer generell substitusjon)

La σ_1 og σ_2 være substitusjoner. Vi sier at σ_2 er **mer generell** enn σ_1 hvis det finnes en substitusjon τ slik at $\sigma_1 = \sigma_2\tau$.

Er $\{f(y)/x\}$ mer generell enn $\{f(a)/x\}$?

Ja, siden $\{f(a)/x\} = \{f(y)/x\}\{a/y\}$.

Er $\{f(a)/x\}$ mer generell enn $\{f(y)/x\}$?

Nei, for det finnes ingen substitusjon σ slik at $\{f(y)/x\} = \{f(a)/x\}\sigma$.

Er $\{f(y)/x\}$ mer generell enn $\{f(y)/x\}$?

Ja, siden $\{f(y)/x\} = \{f(y)/x\}\epsilon$. (Husk: ϵ er identitetssubstitusjonen.)

Definisjon (Unifikator)

La s og t være termer. En substitusjon σ er

- en **unifikator** for s og t hvis $s\sigma = t\sigma$.
- en **mest generell unifikator (mgu)** for s og t hvis
 - den er en unifikator for s og t , og
 - den er mer generell enn alle andre unifikatorer for s og t .

Vi sier at s og t er **unifiserbare** hvis de har en unifikator.

La $s = f(x)$ og $t = f(y)$.

- $\sigma_1 = \{a/x, a/y\}$ er en unifikator for s og t
- $\sigma_2 = \{y/x\}$ og $\sigma_3 = \{x/y\}$ er også unifikatorer for s og t
- σ_2 og σ_3 er de mest generelle unifikatorene for s og t

Variabelomdøping

- Fra det foregående eksempelet ser vi at to termer kan ha flere forskjellige mest generelle unifikatorer.
- Disse mgu-ene er imidlertid like **opp til omdøping av variable**.

Definisjon (Variabelomdøping)

En substitusjon η er en **variabelomdøping** hvis

- 1 $x\eta$ er en variabel for alle $x \in \mathcal{V}$, og
- 2 $x\eta \neq y\eta$ for alle $x, y \in \mathcal{V}$ slik at $x \neq y$.

Er disse substitusjonene variabelomdøpinger?

- $\sigma_1 = \{z/x, x/y, y/z\}$ **Ja.**
- $\sigma_2 = \{z/x, y/z\}$ **Nei, siden $y\sigma_2 = z\sigma_2$.**
- $\sigma_3 = \{z/x, x/y, y/z, a/u\}$ **Nei, siden $u\sigma_3$ ikke er en variabel.**

Unikhet “opp til omdøping av variable”

Påstand

Hvis σ_1 og σ_2 er mest generelle unifikatorer for to termer s og t , så finnes en variableomdøping η slik at $\sigma_1\eta = \sigma_2$.

- Bevis som oppgave?

Deltermer

Definisjon (Deltermer)

Mengden av **deltermer** av en term t er den minste mengden T slik at

- $t \in T$, og
- hvis $f(t_1, \dots, t_n) \in T$, så er hver $t_i \in T$.

Alle termer i T utenom t er **ekte deltermer** av t .

La $s = gx$.

- Deltermer er: x, gx
- Ekte deltermer er: x

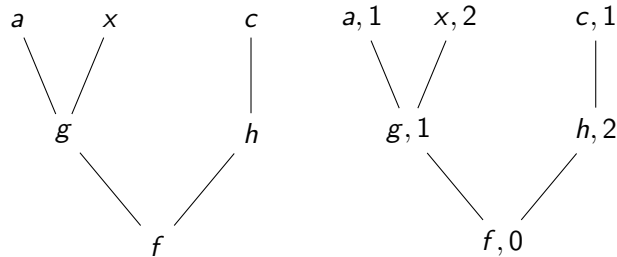
La $t = f(x, a)$.

- Deltermer er: $x, a, f(x, a)$
- Ekte deltermer er: x, a

- En term er altså en delterm av seg selv.

Nummererte termtrær

- Vi har sett at termer kan representeres med trær.
- Når vi unifiserer er det gunstig å nummerere barna til noder i termtræet:



- Slike trær kalles **nummererte termtrær**.
- Vi referer til roten til det nummererte termtræet til en term t som $\text{rot}(t)$.

Kritisk par

- Når vi skal unifisere to termer t_1 og t_2 er vi interessert i å finne par av deltermer som er **ulike**.
- Samtidig er det ønskelig å se på ulike deltermer så nærme roten som mulig.

Definisjon (Kritisk par)

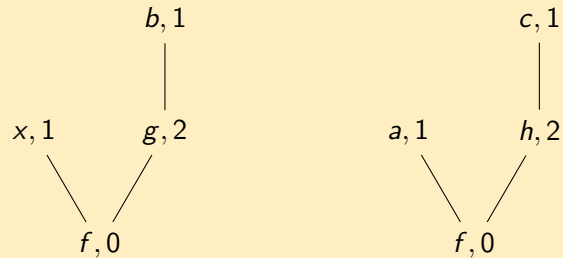
Et **kritisk par** for to termer t_1 og t_2 er et par $\langle k_1, k_2 \rangle$ slik at

- k_1 er en delterm av t_1
- k_2 er en delterm av t_2
- når vi tenker på termer som nummererte termtrær så er
 - $\text{rot}(k_1)$ forskjellig fra $\text{rot}(k_2)$
 - stien fra $\text{rot}(t_1)$ til $\text{rot}(k_1)$ er **lik** stien fra $\text{rot}(t_2)$ til $\text{rot}(k_2)$

- Merk: stiene kan være tomme, dvs. at termene er ulike allerede i rotsymbolet. Tomme stier er trivielt like...

Eksempel

La $s = f(x, gb)$ og $t = f(a, hc)$. Vi får følgende nummererte termtrær:



- Er $\langle b, c \rangle$ kritisk par for s og t ?
 - Nei, stien fra rot(s) til rot(b) er ulik stien fra rot(t) til rot(c).
- Er $\langle x, a \rangle$ kritisk par for s og t ? *Ja*.
- Er $\langle gb, hc \rangle$ kritisk par for s og t ? *Ja*.

Algoritme: unifiser(t_1, t_2)

```

σ := ε;
while (t1σ ≠ t2σ) do
  velg et kritisk par ⟨k1, k2⟩ for t1σ, t2σ;
  if (hverken k1 eller k2 er en variabel) then
    return "ikke unifiserbare";
  end if
  x := den av k1, k2 som er variabel (hvis begge er, så velg én);
  t := den av k1, k2 som ikke er x;
  if (x forekommer i t) then
    return "ikke unifiserbare";
  end if
  σ := σ{x/x};
end while
return σ;
  
```

Egenskaper ved unifiseringsalgoritmen

- Hvis termene t_1 og t_2 er unifiserbare, så returnerer algoritmen en mest generell unifikator for t_1 og t_2 .
- Denne mgu-en er en representant for alle andre unifikatorer for t_1 og t_2 .
- Hvis t_1 og t_2 **ikke** er unifiserbare, så returnerer algoritmen "ikke unifiserbare".