۱۵

جمله بسته (closed formula): جمله اي است که در آن هيچ متغير آزادي وجود نداشته باشد.

مثلا $\exists x\,(p(x,y) \wedge q(x))$ جمله بسته نيست اما $\forall y \exists x\,(p(x,y) \wedge q(x))$ يك جمله بسته است.

بستار عمومی (universal closure) : اگر F یک جمله باشد در این صورت $\forall F$ به این معنی است که روی تمام متغیر ها سور عمومی گرفته شود که به آن بستار عمومی گفته می شود که خود یک جمله بسته است .

* بستار وجودی (existential closure) : اگر F یک جمله باشد در این صورت ∃ F به این معنی است که روی تمام متغیر ها سور وجودی گرفته شود که به آن بستار وجودی گفته می شود که خود یک جمله بسته است .

مثال :

$\boxed{F = q(x) \wedge p(x,y)}$

عموم

$\underline{\forall F} = \forall x \forall y\, (p(x,y) \wedge q(x))$

$\underline{\overset{\exists}{\cancel{\forall}} F} = \overset{\exists}{\cancel{\forall}} x\, \overset{\exists}{\cancel{\forall}} y\, (p(x,y) \wedge q(x))$

وجودی است

(۱) برای بیان مفهوم ابتدا باید تفسیر را بگوییم و برای بیان تفسیر ابتدا باید پیش تفسیر را بگوییم

* مفهوم (semantic) : چگونگی انطباق جملات بر حقایق جهان عینی

* پیش تفسیر (pre-interpretation) : یک پیش تفسیر زبان مرتبه اول L تشکیل شده است از :

a) یک مجموعه غیر تهی به نام دامنه پیش تفسیر (D)

b) به ازای هر ثابت در L ،نسبت دادن یک عضو از D

c) به ازای هر تابع nمتغیر در L ،نسبت دادن یک تابع از

$\boxed{D^n \to D}$

(1)

* تفسیر (interpretation) : یک تفسیر I از زبان مرتبه اول L عبارت است از یک پیش تفسیر J با دامنه D به گونه ای که :

برای هر n-predicate در L ، نسبت دادن تابعی از

$\boxed{D^n \to \{T,F\}}$ اگر یکی دوتا شرایط اضافه کنیم تفسیر می‌شود پیش تفسیر

* مفسر متغیر (variable assignment) : فرض کنید J یک پیش تفسیر از زبان مرتبه اول L باشد یک مفسر متغیر ، تابعی است که به هر متغیر در L ، عنصری را در دامنه J نسبت دهد .

(2)

تعریف : فرض کنید J یک پیش تفسیر از زبان مرتبه اول L با دامنه D باشد V یک مفسر متغیر (تفسیر کننده)باشد یک "مفسر ترم(term-assignment) " به صورت زیر تعریف می گردد:

تفسیر کننده متغیر

1) برای هر متغیر در L ، ارزش آن بر اساس V به دست می آید.

term

2) برای هر ثابت در L ، ارزش آن به وسیله پیش تفسیر به دست می آید .

term

3) اگر $t_1',\ldots, t_n'$ ارزش ترم های $t_1,\ldots, t_n$ باشند و f یک تابع باشد ، آن گاه ارزش آن برابر است با $f'(t_1',\ldots, t_n') \in D$

ارزش f

ارزش است

3

تعریف :فرض کنید I یک تفسیر با دامنه D برای زبان مرتبه اول L و V تابع تفسیر کلمات باشد ، آن گاه به هر جمله در L می توان به شیوه زیر مقدار درست یا غلط را نسبت داد :

a) اگر یک اتم به شکل $p(t_1,\ldots, t_n)$ که $t_1,\ldots, t_n$ ترم هستند داشته باشیم ، آن گاه مقدار درستی آن از طریق $p'(t_1',\ldots, t_n')$ به دست می آید .(که $p'$ و $t_1',\ldots, t_n'$ ارزش p و $t_1,\ldots, t_n$ به وسیله تابع مفسر V است)

b) اگر F و G دو جمله باشند ، آن گاه ارزش جملات $\sim F$ ، $F \wedge G$ ، $F \vee G$ ، $F \rightarrow G$ و $F \leftrightarrow G$ به وسیله جدول درستی به دست می آید

4

(2)

ارزش سورها :

c) اگر F یک جمله باشد ، آن گاه مقدار $\exists x F$ درست است اگر $\exists d \in D$ به طوری که نسبت به تفسیر I و ارزش گذاری V ، مقدار V(x/d) ارزش True داشته باشد . سور وجودی

d) اگر F یک جمله باشد ، آن گاه مقدار $\forall x F$ درست است اگر برای تمام $d \in D$ که نسبت به تفسیر I و ارزش گذاری V ، مقدار V(x/d) ارزش True داشته باشد . سور عمومی

(5)

## جدول درستی :

| F | G | ~F | F∧G | F∨G | F→G | F↔G |
|---|---|---|---|---|---|---|
| true | true | false | true | true | true | true |
| true | false | false | false | true | false | false |
| false | true | true | false | true | true | false |
| false | false | true | false | false | true | true |

(6)

* **تعریف** : فرض کنید I یک تفسیر زبان مرتبه اول L و F یک جمله در L باشد :

- گوئیم F نسبت به I ، ارضاشدنی(satisfiable) است اگر $\exists$ (F) نسبت به I ، درست(True) باشد (سور وجودی اش درست باشد)

برای یکی درست باشد
همه را شامل می شود

- گوئیم F نسبت به I ، معتبر(valid) است اگر( $\forall$ (F نسبت به I، درست(True) باشد
- گوئیم F نسبت به I ، ارضانشدنی(unsatisfiable) است اگر( $\exists$ (F) ~ نسبت به I، درست(True) باشد
- گوئیم F نسبت به I ، غیرمعتبر(non-valid) است اگر (( $\forall$ (F ~ نسبت به I ، درست(True) باشد

7

**نکته** : اگر اتم A به صورت مثبت (یا منفی ) در جمله W ظاهر شود ، آن گاه این اتم در جملات $\forall W$ (برای هر W) ، $\exists W$ (وجود دارد برای W) ، $W \wedge V$ ، $W \vee V$ و $V \leftarrow W$ نیز به صورت مثبت (یا منفی) ظاهر می گردد .

اگر اتم A به صورت مثبت (یا منفی ) در جمله W ظاهر شود ، آن گاه این اتم در جملات $\sim W$ و $V \leftarrow W$ نیز به صورت منفی (یا مثبت) ظاهر می گردد .

W نتیجه می دهد V

اگر اول مثبت ظاهر شد در not آن منفی ظاهر می شود

اگر منفی بود در not به صورت منفی ظاهر می شود.

8

(3)

لیترال می‌تواند هم مثبت باشد و یا هم منفی

Literal : یک لیترال ، یک اتم یا منفی یک اتم است .

Clause: یک Clause جمله ای به صورت

کلیه متغیرها

$\forall x_1 \dots \forall x_s (L_1 \vee \dots \vee L_m)$ است که $L_i$ها در آن "لیترال " و $x_1$ ,...., $x_s$ تمام متغیر هایی هستند که در $L_1 \vee \dots \vee L_m$ وجود دارند و سور عمومی برای همه آن ها وجود دارند.(متغیر آزاد در یک clause وجود ندارد)

چند لیترال

OR ها

روی همه متغیرها

هم سور عمومی گرفتیم

چند مثال از clause :

مثال از لیترال و Clause

$\forall x \forall y \forall z\ (p(x,z) \vee \sim q(x,y) \vee \sim r(y,z))$

$\forall x \forall y\ (\sim p(x,y) \vee \pi(f(x,y),a))$

x و y و z متغیر

روی تمام متغیرها سور عمومی گرفته است

این مجموعه Clause است (9)

نکته : یک سری لیترال ها مثبت هستند یک سری منفی - این ها را باید از هم جدا نمود.

میتوان $\forall x_1 \dots \forall x_s (A_1 \vee \dots \vee A_K \vee \sim B_1 \vee \dots \vee \sim B_n)$ را به صورت $A_1, \dots, A_K \leftarrow B_1, \dots, B_n$ نمایش داد .(به صورتی که تمام متغیر ها دارای سور عمومی هستند و کاما در میان $B_i$ ها به معنی " ^ " و در میان $A_i$ ها به معنی " v " است .

کلیه متغیرها

مثبت

منفی

منفی $\wedge$ =

مثبت $\vee$ ⟸

$\forall x_1 \dots \forall x_s (A_1 \vee \dots \vee A_K \vee \sim B_1 \vee \dots \vee \sim B_n)$ معادل است با :

$\forall x_1 \dots \forall x_s (A_1 \vee \dots \vee A_K \leftarrow B_1 \wedge \dots \wedge B_n)$

توضیح در پشت نوشتم ( p.15 جزوه دست نوشته )

(15)

سور ها را نگرفتیم

clause

* Clause برنامه مشخص( definite program

clause)(DPC): clause ای به فرم $A \leftarrow B_1, \ldots, B_n$ است

(چپ) که در آن A "راس (head)" و $B_1, \ldots, B_n$ "بدنه (body) " (راست)

نامیده می شود .

Clause یکتا (unit clause): DPC ای به فرم

$A \leftarrow$

است (یک DPC با بدنه تهی) . (بدنه اش تهی است)

* نکته : در $A \leftarrow B_1, \ldots, B_n$ اگر تمامی $B_i$ ها درست باشند ، آن گاه A نیز درست است . در $A \leftarrow$ همواره A درست است .

(الزام در جزوه دست نویس)

11

* برنامه مشخص(definite program)(DP):

مجموعه متناهی از DPC هااست .

تعریف :در یک DP، مجموعه تمام DPC ها با راس یکسان (همگی با راس p)، " تعریف p(definition of p) " نامیده می شود. (راس یکسان یعنی همه به یک نتیجه برسند)

هدف مشخص (definite goal)(DG): یک DPCبه

فرم $\leftarrow B_1, \ldots, B_n$ (راس در آن تهی است)

است (عنوان در آن ، تهی است ) که هر یک از $B_i$ ها یک " زیر هدف " نامیده می شود . $\rightarrow B_i$

12

**نکته:** $\leftarrow B_1, \dots, B_n$ معادل است با :

هم لاها

$\forall y_1 \dots \forall y_r (\sim B_1 \vee \dots \vee \sim B_n)$ یا

$\sim\exists y_1 \dots \sim\exists y_r (B_1 \wedge \dots \wedge B_n)$ تبدیل می شود این از نقیض استفاده کردیم

clause تهی (empty clause) : با □ نشان داده می شود که هم بدنه و هم عنوان در آن ،تهی است. این clause به عنوان یک تناقض شناخته می شود .

Horn clause: clause ای است که یا یک definite program است یا یک definite goal می باشد .

13

این در اعداد صحیح صدق نمی کند

که مجموعه اعداد صحیح مدلی برای I نیست (یعنی y ای وجود دارد که از همه x بزرگتر است)

یعنی تفسیر درستی برای این نیست

توضیح

مدل (model) : فرض کنید I یک تفسیر از زبان مرتبه اول L و F یک جمله بسته باشد ، در این صورت I یک مدل برای F است اگر F نسبت به I درست (True) باشد .

**مثال :** جمله $\forall x \exists y\ p(x,y)$ و تفسیر I را به صورت زیر در نظر بگیرید :

فرض کنید دامنه D شامل اعداد صحیح غیر منفی باشد و p نیز رابطه > باشد . در این صورت در I ، جمله فوق بیانگر ایم مسئله است که "برای هر عدد صحیح غیر منفی ، یک عدد صحیح غیر منفی بزرگتر از آن وجود دارد" اما I مدلی برای جمله $\exists y \forall x\ p(x,y)$ نیست .

14

این جمله درست است

y وجود دارد که از همه اعداد بزرگتر است اما این طور نیست ← پس True نیست

در مدل اعداد صحیح

اعداد طبیعی غیر منفی یک مدل است

تعریف : فرض کنید T یک تئوری مرتبه اول و L زبان T باشد .
یک مدل برای T ، یک مفسر برای L است که مدلی برای
هریک از axiom های T می باشد .

اگر T دارای مدلی باشد ، می گوئیم T سازگار است .

تعریف : فرض کنید S مجموعه ای از جملات بسته زبان مرتبه
اول L و I یک تفسیر برای L باشد ، می گوئیم I یک مدل
برای S است اگر I مدلی برای هریک از جملات S باشد .

نکته: اگر $S=\{F_1, ..., F_n\}$ مجموعه ای متناهی از جملات بسته
باشد ، آن گاه I مدلی برای S است اگر و تنها اگر I مدلی
برای $F_1 \wedge ... \wedge F_n$ باشد .

15

تعریف : فرض کنید S مجموعه ای از جملات بسته زبان
مرتبه اول L باشد :

1. گوئیم S ، satisfiable است اگر L ، تفسیری مانند I
داشته باشد که I یک مدل برای S باشد .
2. گوئیم S ، valid است اگر هر تفسیر L ، یک مدل برای
S باشد .
3. گوئیم S ، un satisfiable است اگر هیچ تفسیر L ،
مدلی S برای نباشد .
4. گوئیم S ، nonvalid است اگر L ، تفسیری داشته
باشد که مدلی برای S نباشد .

16

(5)

جلسه پنجم : 93.9.4

* **نتیجه منطقی** : اگر S مجموعه ای از جملات بسته و F یک جمله بسته از زبان مرتبه اول L باشد ، می گوئیم F یک نتیجه منطقی Sاست اگر برای هر تفسیر I از L ، که I مدلی برای S است نتیجه دهد که I مدلی برای F است .

اگر $S=\{F_1,\ldots,F_n\}$ مجموعه ای متناهی از جملات بسته باشد ، آن گاه F یک نتیجه منطقی S است اگر و تنها اگر $F_1 \wedge \ldots \wedge F_n \rightarrow F$ معتبر (valid) باشد .

معتبر = یعنی تحت هر تفسیری True باشد

۱۷

قضیه: اگر S مجموعه ای از جملات بسته و F یک جمله بسته از زبان مرتبه اول L باشد ، آن گاه F یک نتیجه منطقی S است اگر و تنها $S \cup \{\sim F\}$ ، unsatisfiable باشد.

اثبات: فرض کنید F یک نتیجه منطقی S ، I برای هر تفسیری برای L و مدلی برای S باشد ، در این صورت I مدلی برای F نیز هست ، بنابراین I مدلی برای $S \cup \{\sim F\}$ نیست پس $S \cup \{\sim F\}$ ، unsatisfiable است .

وقتی مدلی برای نقیضش نیست پس اجتماع آن با S ارضا نشدنی است

توضیح

وقتی I مدلی برای S هست برای نقیضش نیست

به صورت بر عکس ، فرض کنید $S \cup \{\sim F\}$، unsatisfiable است . فرض کنید I تفسیری دلخواه برای L و مدلی برای S باشد ، از آن جایی که $S \cup \{\sim F\}$، unsatisfiable است، پس I نمی تواند مدلی برای $\sim F$ باشد پس I مدلی برای F است پس F یک نتیجه منطقی S است

۱۸

* مثال : فرض کنید $S=\{ p(a) , \forall x (p(x)\rightarrow q(x)) \}$ و $F=q(a)$

، در این صورت F یک نتیجه منطقی S است

یعنی هم چیز

* زیرا : اگر I مدلی برای S باشد ،p(a) با تفسیر I درست است و درست بودن $\forall x (p(x)\rightarrow q(x))$ ، درست بودن $p(a)\rightarrow q(a)$ را نتیجه می دهد ، در نهایت هم از درست بودن p(a) و $p(a)\rightarrow q(a)$ ، می توان نتیجه گرفت که q(a) درست است .

True است

پس

درست است

q(a) نتیجه لیری منطقی S است

19

کلمه ثابت

* تعریف : ترمی که شامل متغیر نباشد ، ground term (کلمه ثابت)نامیده می شوند . به صورت مشابه اتمی که شامل متغیر نباشد ، ground atom (اتم ثابت) نامیده می شود .

* تعریف : فرض کنید L یک زبان مرتبه اول باشد ، Herbrand universe ($U_L$) برای L ، مجموعه تمام کلمات ثابتی است که می توانند از ترکیب ثابت ها و توابع داخل L، ساخته شوند .

مثال :برنامه زیر را در نظر بگیرید :

$$\begin{cases} P(x)\leftarrow q(f(x),g(x)) \\ R(y)\leftarrow \end{cases}$$

در این صورت $U_L$ برای L به صورت زیر است :

{ a , f(a) , g(a) , f(f(a)) , f(g(a)) , g(f(a)) , g(g(a)) , ... }

تابع ها f و g ، و اثر تمام ثابت ها بر روی توابع

20

herbanduniverse

6

تعریف : ترمی که شامل متغیر نباشد ، ground term (کلمه ثابت)نامیده می شوند . به صورت مشابه اتمی که شامل متغیر نباشد ، ground atom (اتم ثابت) نامیده می شود .

تعریف : فرض کنید L یک زبان مرتبه اول باشد ، Herbrand universe ($U_L$) برای L ، مجموعه تمام کلمات ثابتی است که می توانند از ترکیب ثابت ها و توابع داخل L، ساخته شوند .

مثال :برنامه زیر را در نظر بگیرید :

P(x)←q(f(x),g(x))

R(y) ←

در این صورت $U_L$ برای L به صورت زیر است :

{ a , f(a) , g(a) , f(f(a)) , f(g(a)) , g(f(a)) , g(g(a)) , ... }

21

تعریف : زبان مرتبه اول L را در نظر بگیرید Herband base ($B_L$) برای L ، مجموعه تمام کلمات ثابتی است که به وسیله predicate ها به دست می آیند. p و q دوتا

مثال :برنامه زیر را در نظر بگیرید :

P(x)←q(f(x),g(x))

R(y) ←

در این صورت $B_L$ برای L به صورت زیر است :

{ p(a) , q(a,a) , r(a) , p(f(a)) , p(g(a)) , q(a,f(a)) , q(f(a),a) , ... }

دوتا ثابت

22 اینها herband base است

تعریف: زبان مرتبه اول $L$ را در نظر بگیرید، پیش تفسیر herband (herband-preinterpratation) برای $L$، پیش تفسیری با خصوصیات زیر است:

تفسیر Herbrand (Herbrand interpretation):

تفسیر Herbrand برای L، تفسیری است که بر اساس پیش تفسیر Herbrand به دست می آید.

مدل، تعریف: فرض کنید L یک زبان مرتبه اول و S مجموعه ای از جملات بسته در L باشد، یک مدل Herbrand (Herbrand model) برای S، یک تفسیر Herbrand برای L است به گونه ای که مدلی برای S باشد.

Herbrand universe و Herbrand base برای S را به ترتیب با $U_S$ و $B_S$ نشان می دهیم.

23

a) دامنه این پیش تفسیر، universe herband ($U_L$) است

b) ثابت های در $L$، به خودشان نسبت داده می شوند.

(2) قضیه: اگر S مجموعه ای از clause ها و دارای یک مدل باشد، آن گاه S دارای یک مدل Herbrand می باشد.

اثبات: فرض کنید I تفسیری برای S باشد، ما تفسیر Herbrand (I′) برای S را به صورت زیر تعریف می کنیم:

ثابت کنیم I′ درست باشد

$$I' = \{\, p(t_1,\ldots,t_n) \in B_S : p(t_1,\ldots,t_n) \text{ is true wrt } I \,\}$$

واضح است که اگر I یک مدل باشد، آن گاه I′ نیز یک مدل Hmodel است.

24

c) اگر F یک تابع n متغیره در $L$ باشد، آن گاه تابع n تایی $(U_L)^n \to U_L$ را می توان به صورت زیر $(t_1,\ldots,t_n) \to f(t_1,\ldots,t_n)$ تعریف کرد.

(7)

قضیه 3: اگر S مجموعه ای از clause ها باشد ، آن گاه S ، unsatisfiable است اگر و تنها اگر S هیچ مدلHerbrand ای نداشته باشد .

اثبات: اگر S ، satisfiable باشد ، طبق قضیه 2 دارای یک مدل Herbrand است .

نکته : با در نظر نگرفتن محدودیت این که S باید مجموعه ای از clause ها باشد ، قضایای 2 و 3 برقرار نخواهند بود . به عبارت دیگر اگر S مجموعه ای از جملات بسته اختیاری باشد ، آن گاه در حالت کلی با استفاده از تفسیر های Herbrand نمی توان گفت که unsatisfiable است .

25

تعریف : دو جمله W و V ، به صورت منطقی برابر(logically equivalent) هستند اگر $\forall\,(W \leftrightarrow V)$ معتبر ( valid) باشد .

26

ادامه از جلسه پنجم : امتحان این درس ② بخش دارد یکی بخش منطقی که این حالت ① و دیگری بخش برنامه نویسی. در بخش منطقی مثل مثال های اسلایدهای قبل

به عنوان مثال اسلاید
اسلاید ~~20~~ قبل
و یا اسلاید 19

# برنامه نویسی منطق

## مبحث سوم

دکتر محمد ابراهیم شیری

1

* termها یا ثابت هستند (ground) و یا متغیر هستند.

# همسانسازی (UNIFICATION)

تعریف : یک تابع جایگذاری (substitution) $\theta$، یک مجموعه متناهی از $\{ V_1/t_1 , \ldots , V_n/t_n \}$ است که هر $V_i$ یک متغیر و هر $t_i$ ترمی متمایز با $V_i$ است . در ضمن متغیر های $V_1,\ldots, V_n$ نیز متمایز هستند . هر $V_i/t_i$ یک الزام ( binding) برای $V_i$نامیده می شود. (به جای $V_1 \leftarrow t_1$ و به جای $V_n$، $t_n$ گذاشتیم.)

الزام

نکته : اگر تمام $t_i$ ها ترم های ثابت (ground term) باشند ، $\theta$یک جایگذاری ثابت (ground substitution)نامیده می شود و در صورتی که تمام $t_i$ ها متغیر باشند ، $\theta$یک جایگذاری متغیر محض(variable-pure substitution)نامیده می شود

2

ادامه از جلسه پنجم : اسلاید ② اسلاید سوم :

$buf(bob)$

$pig(wilber)$

$$\forall x \forall y \; buffalo(x) \wedge pig(y) \longrightarrow Fasterthan(x,y)$$

S بشود

می‌خواهیم از این برسیم به اینکه بگوییم

$Faster(bob, wilber)$

الان اینجا از تابع جایگذاری استفاده می‌کنیم ؛ یک شبه گزاره منطقی را به ما می‌دهد.

می‌خواهیم ببینیم این تابع جایگذاری به چه دردی می‌خورد؟

---

$$\theta = \{x/bob \,,\, y/wilber\}$$ این یک تابع جایگذاری است

برای تابع جایگذاری به صورت زیر می‌نویسیم :

$$F\theta \Rightarrow bufallo(bob) \wedge pig(wilber) \longrightarrow Fasterthan(bob, wilber)$$

کلا تابع جایگذاری را به این صورت می‌نویسیم

$$\boxed{\forall x \forall y \; buF(x) \wedge pig(y) \rightarrow fasterthan(x,y) \longleftarrow F}$$

---

دقیقه ۴۷ :

* حل مثال : $\theta = \{x/f(y) \,,\, y/x\}$ $\quad \sigma = \{x/a \,,\, y/b \,,\, z/y\}$

به جای x ها a و به جای y ها b و به جای z ها y می‌گذاریم

$$\theta\sigma = \{x/(f(y)\sigma) \,,\, y/(x\sigma) \,,\, x/a \,,\, y/b \,,\, z/y\}$$

آنهایی که صورت یکی دارند حذف می‌شوند

$$\theta\sigma = \{x/f(b) \,,\, y/a \,,\, \cancel{x/a} \,,\, \cancel{y/b} \,,\, z/y\}$$ حالا صورت و مخرج ها حذف می‌شوند

$\theta\sigma = \{a/f(b), y/a, z/y\}$ اسلاید یکی کم دارد.

---

مثال برای همسان ساز:

($a$ ثابت است)

$S_1 = \{x, f(y)\}$

$S_2 = \{z, f(y)\}$

$\theta = \{x/a, z/a\}$

? $S_1\theta = \{a, f(y)\}$ و $S_2\theta = \{a, f(y)\}$

این دوتا برابر شدند یعنی همسان شدند به همین علت می‌گوییم $\theta$ همسان ساز $S_1$ و $S_2$ است با جاگذاری می‌توان این دو را یکی کرد. در اثبات قضایا خیلی به درد می‌خورد. ($y$ متغیر آزاد است)
$\theta$ تابع جایگذاری است. (این مورد در اثبات قضایا بسیار کاربرد دارد)

(2)

تعریف : یک عبارت (expression) یا یک ترم یا لیترال یا conjunction یا disjunction لیترال ها می باشد.

$\vee \; \neg \; \wedge$

یک عبارت ساده (simple expression) یا یک ترم است یا یک اتم .

در اسلاید زیر یعنی به جای $V_1$ ، $t_1$ می گذاریم . .

تعریف : فرض کنید $\theta = \{ V_1/t_1 , \ldots , V_n/t_n \}$ یک تابع جایگذاری و E یک عبارت باشد در این صورت $E\theta$ (نمونه E تحت $\theta$)، عبارتی است که از روی E با جایگذاری $t_i$ ها به جای $V_i$ ها به دست می آید .

نکته : اگر $E\theta$ ثابت باشد ، آن گاه $E\theta$ یک نمونه عبارت ثابت (ground instance) E نامیده می شود .

مثال : فرض کنید $E=p(x,y,f(a))$ و $\theta = \{ x/b , y/x \}$ در این صورت $E\theta = p(b,x,f(a))$

هم راهی نویسیم در $u_i$ ها اونی که صورت و مخرجش برابر است را حذف می کنیم و در قسمت ۲ جا هایی که ۲ چپ ها در قسمت اول است . صورتش را حذف می کنیم .

* **نکته** : اگر $S = \{ E_1 , \dots , E_n \}$ یک مجموعه متناهی از عبارت ها و $\theta$ یک تابع جایگذاری باشد آن گاه

$\boxed{S\theta = \{ E_1 \theta, \dots , E_n \theta\}}$ روی تک تک عبارات تابع جایگذاری عمل شود

* **تعریف** : فرض کنید $\theta = \{ u_1/s_1 , \dots , u_m/s_m \}$ و $\sigma=\{ V_1/t_1 , \dots , V_n/t_n \}$ دو تابع جایگذاری باشند در این صورت ، ترکیب $\theta\sigma$ تابع جایگذاری ای است که از مجموعه $\{u_1/s_1 \sigma , \dots , u_m/s_m \sigma , V_1/t_1 , \dots , V_n/t_n \}$ به وسیله حذف $u_i/s_i \sigma$ هایی که $u_i = s_i \sigma$ و حذف $V_j/t_j$ ها که $V_j \in \{u_1 , \dots , u_m \}$ به دست می آید .

۲ در قسمت ($u_1, \dots, u_m$) است حذف می شود .

* s می تواند ثابت یا متغیر باشد

ترکیب دو تابع جایگذاری هم جایگذاری است ($\theta\sigma$ یک تابع جایگذاری است)

* نحوه به دست آوردن تابع جایگذاری :

به جای اینکه مخرج را بنویسیم $S_1$ می نویسیم $S_1\sigma$ چون می خواهیم $\theta\sigma$ را بنویسیم

**مثال** : فرض کنید $\theta = \{ x/f(y) , y/x \}$ و $\sigma = \{ x/a , y/b , z/y \}$ در این صورت :

$\boxed{\theta\sigma} =\{ x/f(b) , z/y \}$ f(y) به جای x

b به جای y و ...

اثر ترکیب دو تابع بالا می شود

چون اثر کرده است $u_i/s_i$ هایی که صورت و مخرجشان برابر است حذف می شود.

* حل در برگه دست نویس :

**تعریف** : یک جانشینی با مجموعه تهی ،جایگذاری یکانی( identity substitution )($\varepsilon$) نامیده می شود .

* تابعی که کار خاصی نمی کند و هر عبارتی را به خودش می برد

**نکته** : برای تمام عبارت های E داریم : $E\varepsilon = E$ با اپسیلون نشان می دهم

(3)

(a) هر تابع جایگذاری اثرش با تابع یکتائی می‌شود خودش.

**قضیه:** فرض کنید $\theta$ ، $\sigma$ و $\gamma$ سه تابع جایگذاری باشند در این صورت :

a) $\theta\varepsilon = \varepsilon\theta = \theta$

اگر اول $\theta$ روی عمل کند و بعد $\varepsilon$ را معادل این است بنویسیم

b) برای تمام عبارت های E داریم : $(E\theta)\sigma = E(\theta\sigma)$

$\theta\sigma$ را روی E عمل کنیم

c) $(\theta\sigma)\gamma = \theta(\sigma\gamma)$ خاصیت ترکیب پذیری دارند فرقی نمی‌کند کدام یک با هم ترکیب شوند.

اثبات قسمت a $(\theta\varepsilon = \varepsilon\theta = \theta)$:

با توجه به تعریف $\varepsilon$ ، واضح است .

اگر هر $\theta$ یعنی کاری نمی‌کند . با توجه به تعریف اپسیلون

---

اثبات قسمت دوم : ابتدا برای یک متغیره اثبات می‌کنیم بعد تعمیم می‌دهیم

(b) فرض می‌کنیم x معادل یکی از $v_i$ ها باشد $\theta$ می‌شود خودش. متعلق به $v_1$ تا $v_n$ باشد نه $u_1$ تا $u_m$ . فرض کنید x معادل یکی از $u_i$ ها باشد . برابر است وقتی معادل $u_i$ باشد و $\theta$ روی آن اثر کند . $\theta$ فقط $v_1$ تا $v_n$ را تغییر نمی‌دهد ، $u_1$ تا $u_m$ را تغییر می‌دهد و به جایش $s$ 

اثبات قسمت b $((E\theta)\sigma = E(\theta\sigma))$: $\theta$ که تغییر می‌دهد و به جایش $t$ می‌گذارد

کافی است قضیه را در حالتی که E یک متغیر ، مثلا x ، است ثابت کنیم . فرض کنید $\theta = \{u_1/s_1, \ldots, u_m/s_m\}$ و $\sigma = \{v_1/t_1, \ldots, v_n/t_n\}$ :

این مربوط به حالت (1) است

(1) اگر $x \notin \{u_1, \ldots, u_m\} \cup \{v_1, \ldots, v_n\}$ آن گاه

ابتدا فرض می‌کنیم که هیچ یک از حالات فوق نباشد $\boxed{(x\theta)\sigma = x = x(\theta\sigma)}$

(2) اگر $x \in \{u_1, \ldots, u_m\}$ ، یعنی $x = u_i$ ، آن گاه پس اگر $\theta$ اثر کند خودش را می‌دهد و اگر $\sigma$ هم اثر کند خودش را می‌دهد

$\boxed{(x\theta)\sigma = s_i\sigma = x(\theta\sigma)}$ ترکیب این دو هم اثر کند خودش را می‌دهد

(3) اگر $x \in \{v_1, \ldots, v_n\} \setminus \{u_1, \ldots, u_m\}$ ، یعنی $x = v_j$ آن گاه

در ترکیب خودش را می‌دهد و فقط به جای مخرج $\theta\sigma$ ، $\sigma$ می‌نویسیم $\boxed{(x\theta)\sigma = t_j = x(\theta\sigma)}$ ← حالت دوم

دقیقه 57

آیا برای یک اثبات کردن برای چند تا هم قابل بسط است مثلاً $\theta\sigma$ یک تابع جایگزین باشد
آن وقت می‌توان آن را با $\lambda$ جایگزین کرد. اول تابع $\sigma$ روی $\theta$ اثر کند بعد $\gamma$ $((\theta\sigma)\gamma)$.
ترکیب روی — می‌نویسیم

اثبات قسمت c $((\theta\sigma)\gamma = \theta(\sigma\gamma))$:

کافی است نشان دهیم اگر x یک متغیر باشد ، آن گاه

$$x((\theta\sigma)\gamma) = x(\theta(\sigma\gamma))$$

در حقیقت با توجه به قسمت b داریم:

$$x((\theta\sigma)\gamma) = (x(\theta\sigma))\gamma = ((x\theta)\sigma)\gamma = (x\theta)\ (\sigma\gamma) = x(\theta(\sigma\gamma))$$

حالا اول $\theta$ اعمال شود بعد $\lambda$.

مثال: بجای x ، f(y) بگذاریم و بجای y ، z بگذاریم.
ابتدا اثر $\theta$ روی عبارت. یعنی هر y را بجای f(x) قرار دهیم

مثال : فرض کنید $\theta = \{ x/f(y)\ ,\ y/z \}$ و

$\sigma = \{ x/a\ ,z/b\}$ در این صورت :

$\theta\sigma = \{ x/f(y)\ ,\ y/b\ ,\ z/b \}$ ترکیب این دو:

فرض کنید $E = p(x\ ,\ y\ ,\ g(z))$ در این صورت :

ابتدا $\theta$ یک اثر می‌دهیم:

$E\theta = p(f(y)\ ,\ z\ ,\ g(z))$ چرا اینجا g(z) تغییر نکرد

$(E\theta)\sigma = p(f(y)\ ,\ b\ ,\ g(b))$ اینجا اثر $\sigma$ ← ترکیب این دو

$E(\theta\sigma) = p(f(y)\ ,\ b\ ,\ g(b)) = (E\theta)\sigma$

ترکیب $\theta\sigma$ روی E بجای x ، f(y) می‌گذاریم. هرجای y داریم
b می‌گذاریم. هرجا z داریم b می‌گذاریم. پس هردو با هم برابر می‌شود.
خیلی راحت چرا؟ یا راحت بود؟ آیا این کار خوب است؟

تعریف : اگر E و F دو عبارت باشند ، می گوئیم E و F ، بدل (variant) همدیگرهستند اگر دو تابع جایگذاری θ و σ وجود داشته باشد به گونه ای که

$\boxed{F=E\sigma \text{ و } E=F\theta}$ که در این صورت این دو بدل هم هستند

همچنین می گوئیم E یک بدل از F یا F یک بدل از E است .

مثال : p(f(x,y) , g(z) , a) یک بدل از p(f(y,x) , g(u) , a) است اما p(x,x) یک بدل از p(x,y) نیست . * چون یک تابع جایگذاری است می‌شود p را به آن می‌توان به آن تبدیل کرد. به جای هر یک از اینها می‌توان دیگری را گذاشت.

در حالت دوم اگر به جای x بگذاریم y دیگر نمی‌توان به جای y گذاشت x و تابع جایگذاری برای آن یافت نمی‌شود *

تعریف درست : اگر E یک عبارت باشد آنگاه تابع جایگذاری که به جای

تعریف : ~~اگر E یک تابع جایگذاری و V مجموعه متغیر های E باشد~~ ، آن گاه یک تابع نامگذاری متغیرها، متغیر دیگر (renaming substitution) برای E ، یک تابع بگذارد تابع جایگذاری جایگذاری متغیر خالص $\{x_1/y_1 , \ldots , x_n/y_n\}$ است درست

به گونه ای که :

متغیر خالص است renaming یا

- $\{x_1 , \ldots , x_n\} \subseteq V$
- $y_i$ ها متمایز هستند
- $(V \setminus \{x_1 , \ldots , x_n\}) \cap \{y_1 , \ldots , y_n\} = \emptyset$

کرده یعنی x برده قبلا به جایش y گذاشته ثابت نمی‌گذارد.

بدل یکدیگر یعنی یکی از روی دیگری بدست آمده است

قضیه : اگر E و F دو عبارت باشند که بدل یکدیگر هستند. آن گاه دو تابع جایگذاری $\theta$ و $\sigma$ وجود دارد به گونه ای که $E=F\theta$ و $F=E\sigma$ که $\theta$ یک تابع نامگذاری برای F و $\sigma$ یک تابع نامگذاری برای E است . (یعنی متغیرها را جا به جا کرده است)

**اثبات** : از آن جایی که E و F ، بدل یکدیگر هستند پس دو جانشینی $\theta_1$ و $\sigma_1$ وجود دارند به گونه ای که $E= F\theta_1$ و $F=E\,\sigma_1$ . فرض کنید V مجموعه متغیر های E و $\sigma$ جانشینی به دست آمده به وسیله $\sigma_1$ با حذف تمام binding های به فرم x/t که $x \notin V$ ، باشد . واضح است که $F=E\sigma$ . علاوه بر این $E=F\theta_1 =E\sigma\theta_1$ . پس در نتیجه $\sigma$ یک تابع نامگذاری برای E است (و اثباتش را نگفت!)

یا به عبارتی تابعی یا ضمیمه به هر دو روی $\theta$ اثر کرد یکی شد به آن می گوییم همسان ساز unifier . یک تابع جایگذاری که وقتی روی $S_1$ اثر کرد و اثر کرد روی $S_2$ این دو با هم یکی شدند.

تعریف : فرض کنید $S_1$ و $S_2$ دو مجموعه از عبارات متناهی و $\theta$ یک تابع جایگذاری باشد . اگر $S_1\theta=S_2\theta$ باشد ، آن گاه $\theta$ را یک **همسان ساز** (unifier) $S_1$ و $S_2$ می نامیم . (در برگه دست نویس توضیح نوشتم)

**نکته** : همسان ساز $\theta$ ممکن است دارای متغیر های آزاد باشد . همسان ساز ممکن است همه متغیرها را جا به جا نکند بلکه بعضی از متغیرها را جا به جا کند

(5) لزومی ندارد s و t یک عبارت باشند ممکن است مجموعه‌ای از عبارات باشند

θ عمومی‌تر از σ است. چون σ ترکیبی از دو تا ست یعنی خیلی متغیرهای بیشتری را جایگذاری می‌کند ولی θ که نمی‌کند چون θ و σ یعنی هم جایگذاری θ است و هم γ پس θ عمومی‌تر است.

**تعریف** : فرض کنید θ و σ دو همسان ساز S و t باشند. اگر چنین دسری پیدا نشود اگر $\sigma=\theta\gamma$ که $\gamma$ هم یک همسان ساز ( تابع جایگذاری )

است . در این صورت می گوئیم θ عمومی تر از σ است

. هم چنین $S\sigma$ را یک نمونه از $S\theta$ می نامیم .

**تعریف** : تابع جایگذاری θ را عمومی ترین همسان ساز

S و t ((mgu) most general unifier) گویند اگر θ یک

همسان ساز S و t باشد و عمومی تر از هر همسان

ساز دیگری باشد .

در اسلاید زیر دو عبارت داریم: برای دو عبارت زیر هیچ تابع جایگذاری نداریم که آنها را unify کند (قابل جایگذاری نیستند) چون اگر بخواهیم f(w) را به y ببریم آرگومان دوم ثابت است و متغیر نیست. چون متغیرها را جایگذاری می‌کرد و نه ثابت‌ها را. یعنی می‌خواهیم یکی را به دیگری ببریم نمی‌شود. حالا دیگری را نگاه کنید.

* مثال : { p(f(x),a) , p(y,f(w)) } unifiable نیست

زیرا آرگومان های دوم نمی توانند unify شوند .

* مثال : { p(f(x),z) , p(y,a) } unifiable است زیرا

σ={ y/f(a) , x/a , z/a } یک unifier است .

mgu برابر است با : θ={ y/f(x) , z/a } تست کنید ببینید

جلسه ششم 93.9.11.

در مثال زیر هر سه عبارت f(x) را دارند ـ دوتا h(y) دارند و دیگری z دارد و در مولفه سوم هر عبارت دوتا a دارند و دیگری b (الفبا ثابت‌ها

تعریف : فرض کنید S مجموعه ای متناهی از عبارات ساده باشد مجموعه ناسازگار (disagreement) S به صورت زیر تعریف می گردد :

کاری نداریم

مجموعه ای است که در آن (از سمت چپ به بعد) علامت هایی را انتخاب کنید که همه عبارت های S آن ، علامت ندارند.

مثال : فرض کنید S دارای سه عبارت می باشد.

S={p(f(x),h(y),a),p(f(x),z,a),p(f(x),h(y),b)}

در این صورت مجموعه disagreement برابر است با:

نداریم

با متغیرها کار داریم و کاری به ثابت‌ها { h(y) , z }



تعریف : فرض کنید S مجموعه ای متناهی از عبارات ساده باشد یك تابع جایگذاري θ را یك همسانساز Unifier براي S نامیم اگر Sθ یگانه (منحصر بفرد) نه باشد

همسانساز θ را عمومي ترین همسانساز گوئیم اگر براي یك همسانساز δ وλ داشته باشي

δ= θ λ

(b) الگوریتم زیر طریقه بدست آوردن جایگشت همسان ساز را بیان می کوید.
حالا طریقه اجرا را با یک مثال می بینیم

# UNIFICATION ALGORithm

اگر $S\sigma_k$ بود استاپ می شود.

همسان سازهای

در غیر این صورت

یعنی کاری نمی کند

1. Put $k=0$ and $\sigma_0=\varepsilon$.

مجموعه $D_k$ را بدست می آوریم disagreement

یک باشد

یگانه باشد

2. If $S\sigma_k$ is a singleton, then stop; $\sigma_k$ is an mgu of S. Otherwise, find the disagreement set $D_k$ of $S\sigma_k$.

→ most general unifier است

(یعنی تفاوت این دو را بدست می آوریم)

3. If there exist v and t in $D_k$ such that v is a variable that does not occur in t,

تابع جایگذاری

به این صورت

تعریف می شود

then put $\sigma_{k+1} = \sigma_k\{v/t\}$, increment k and go to 2. Otherwise, stop; S is not unifiable.

بجای v t را قرار می دهیم

هر بار یک مرحله جلو می رویم تا بالاخره متغیرها با هم همسان شوند
در این صورت عمومی ترین همسان ساز را بدست می آوریم.

به مثال زیر دقت کنید. می خواهیم ببینیم همسان ساز پیدا می شود یا خیر

$D_0$: یعنی اختلاف بعد از اثر دادن

مثال 1: اگر $S = \{ p(f(a),g(x)), p(y,y) \}$

تابع همسان سازهای

a) $\sigma_0 = \varepsilon \longrightarrow$ از این شروع می کنیم

بدست می آید b) $D_0 = \{ f(a) , y \}$ , $\sigma_1 = \{ y/f(a) \}$ , اثر این $D_0$ است روی S

$S\sigma_1 = \{ p(f(a),g(x)) , p(f(a),f(a)) \}$ → $g(a)$ و $f(a)$ با هم تفاوت دارند

c) $\boxed{D_1 = \{ g(x) , f(a) \}}$ بعد از اینکه اثر دادیم می شود یکی نشوند

اختلاف این دو. یعنی بعد از اثر دادن می شود اختلاف این ها. اگر اختلاف

پس S ، unifiable نیست. چون تک عنصری یا single نشد

این $D_0$ را نگاه کنیم. تابع همانی ضرورش است. اختلاف این دو جمله در

اولی $f(a)$ داریم دومی y داریم. اگر بجای y $f(a)$ قرار دهیم

یکی می شود. تا این قسمت یکی می شود حالا اثر $\sigma_1$ روی S. هر جایی
y است a قرار می دهیم $S\sigma_1$ بدست می آید

$\sigma_0$ نگاشت یا جایگشت اولیه را $\sigma_0$ یا تابع همانی می‌گیریم و اولین D را بدست می‌آوریم، بعد به جای z ، a قرار می‌دهیم $\sigma_1$ ـ تا قسمت اول یکسان می‌شود. حالا اثر $\sigma_1$ را روی S بررسی می‌کنیم می‌شود $S\sigma_1$. حالا $D_1$ را پیدا می‌کنیم. $\sigma_2$

مثال 2: اگر $S = \{ p(a,x,h(g(z))), p(z,h(y),h(y)) \}$

a) $\sigma_0 = \varepsilon$ به جای x h(y) قرار می‌دهیم

b) $D_0 = \{ a , z \}$ , $\sigma_1 = \{ z/a \}$ , اختلاف بعدی

$S\sigma_1 = \{ p(a,x,h(g(a))) , p(a,h(y),h(y)) \}$ h(y) و

c) $D_1 = \{ x , h(y) \}$ , $\sigma_2 = \{ z/a , x/h(y) \}$ , h(g(a))

$S\sigma_2 = \{ p(a,h(y),h(g(a))) , p(a,h(y),h(y)) \}$ است

d) $D_2 = \{ y , g(a) \}$ , $\sigma_3 = \{ z/a , x/h(g(a)) , y/g(a) \}$

, $S\sigma_3 = \{ p(a,h(g(a)),h(g(a))) \}$ پس g(a) به جای y می‌گذاریم.

پس S ، unifiable و $\sigma_3$ ، mgu است. اثر $\sigma_3$ روی S singleton شد فقط یکی دارد. فقط یک عبارت شد در ابتدا دو عبارت بود.

در مثال پایین نمی‌شود به جای y دو بار چیزی قرار دهیم به همین علت در آخر نمی‌شود جایگذاری دیگری انجام داد.

مثال 3: اگر $S = \{ p(x,x), p(y,f(y)) \}$

a) $\sigma_0 = \varepsilon$

b) $D_0 = \{ x , y \}$ , $\sigma_1 = \{ x/y \}$ ,

$S\sigma_1 = \{ p(y,y) , p(y,f(y)) \}$

c) $D_1 = \{ y , f(y) \}$

از آن جایی که y در f(y) آمده است. پس S ، unifiable است.

آیا نمی‌شود دو تا متغیر را به جای همدیگر گذاشت

توجه: به جای ثابت‌ها متغیر قرار می‌دهیم مثلاً g(x)/g(a)

سؤال خودم ؟!

(دقیقه 20)