



هوش مصنوعی

حسین کارشناس

دانشکده ریاضی

ترم اول ۹۴ - ۹۳

استنتاج در منطق درجه اول

استنتاج در منطق درجه اول

- چگونه می‌توان رابطه التزام بین جملات در زبان منطق درجه اول را بررسی کرد؟
- رویکرد اول: تبدیل FOL-KB به PL-KB
- تمام جملات KB به جملات معادل در زبان منطق گزاره‌ای تبدیل می‌شود
 - نیاز به تبدیل سورها
- از روش‌های استنتاج در منطق گزاره‌ای استفاده می‌شود
 - الگوریتم بررسی مدل
 - الگوریتم‌های زنجیره‌بندی جلورو و عقب‌رو
 - الگوریتم رفع
- به این روش گزاره‌ای‌سازی (propositionalization) گفته می‌شود

استنتاج در منطق درجه اول

• قوانین تبدیل سورها

• نمونه گذاری عمومی (universal instantiation – UI)

$$\frac{\forall v \alpha}{\text{SUBST}(\{v/g\}, \alpha)}$$

- هر جمله حاصل از جایگزینی یک متغیر (v) با یک واژه ی بنیادی (ground term) – مانند g – قابل استنتاج است

• مثال: $\forall x \text{ King}(x) \wedge \text{Greedy}(x) \Rightarrow \text{Evil}(x)$

با جایگزینی های $\{x/\text{John}\} \quad \{x/\text{Richard}\} \quad \{x/\text{Father}(\text{John})\} \quad \dots$

$\text{King}(\text{John}) \wedge \text{Greedy}(\text{John}) \Rightarrow \text{Evil}(\text{John})$

$\text{King}(\text{Richard}) \wedge \text{Greedy}(\text{Richard}) \Rightarrow \text{Evil}(\text{Richard})$

$\text{King}(\text{Father}(\text{John})) \wedge \text{Greedy}(\text{Father}(\text{John})) \Rightarrow \text{Evil}(\text{Father}(\text{John}))$

$\vdots$

استنتاج در منطق درجه اول

• قوانین تبدیل سورها (ادامه)

• نمونه گذاری وجودی (EI – existential instantiation)

$$\frac{\exists v \alpha}{\text{SUBST}(\{v/k\}, \alpha)}$$

• جمله حاصل از جایگزینی یک متغیر (v) با یک نماد **ثابت** جدید که در KB موجود نباشد (k) قابل استنتاج است

• این نماد یک ثابت Skolem نامیده می شود

• مثال: $\exists x \text{Crown}(x) \wedge \text{OnHead}(x, \text{John})$

$\text{Crown}(C_1) \wedge \text{OnHead}(C_1, \text{John})$

• پس از اعمال نمونه گذاری وجودی می توان جمله حاوی سور وجودی را حذف کرد

• KB حاصل از لحاظ استنتاج با KB اصلی برابر است (inferentially equivalent)

• اگر KB اصلی ارضاء پذیر باشد، KB حاصل از حذف سور وجودی هم ارضاء پذیر است

استنتاج در منطق درجه اول

- گزاره‌ای‌سازی یک FOL-KB

- هر سور وجودی با اعمال نمونه‌گذاری وجودی توسط یک جمله جایگزین می‌شود
- هر سور عمومی با اعمال نمونه‌گذاری عمومی توسط تمام جملات ممکن (حاصل از تمام واژه‌های بنیادی متفاوت موجود در KB) جایگزین می‌شود
- اگر KB شامل تابع باشد تعداد جملات ممکن نامتناهی است

مثال: $Father(Father(Father(John)))$

- تئوری Herbrand: اگر جمله‌ای قابل التزام از یک FOL-KB باشد برای این التزام اثباتی با استفاده از زیرمجموعه‌ای متناهی از جملات در KB گزاره‌ای‌سازی شده وجود دارد

• ساخت افزایشی معادل گزاره‌ای یک FOL-KB

استنتاج در منطق درجه اول

- گزاره‌ای‌سازی یک FOL-KB
- این روش کامل است
- اما در صورت وجود توابع، تصمیم‌گیری در مورد عدم التزام یک جمله از KB ممکن پذیر نیست
- بررسی التزام در منطق درجه اول نیمه‌تصمیم‌پذیر (semidecidable) است
- در فرآیند گزاره‌ای‌سازی تعداد زیادی جملات نامرتبط تولید می‌شود که در بررسی التزام هیچ اثری ندارد
- رویکرد دوم: ترفیع (lifting) و یکسان‌سازی (unification)
- قانون شیوه تصدیق (Modus Ponens) برای بکارگیری در FOL
- قانون رفع (resolution) در FOL

استنتاج در منطق درجه اول

- قانون شیوه تصدیق کلی

- اگر برای جملات پایه p_i و p'_i و q جایگزینی θ وجود داشته باشد طوری که برای هر i داشته باشیم: $\text{Subst}(\theta, p'_i) = \text{Subst}(\theta, p_i)$ آنگاه

$$\frac{p_1', p_2', \dots, p_n', (p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q)}{\text{SUBST}(\theta, q)}$$

- اگر بتوان یک جایگزینی (θ) برای متغیرهای موجود در قسمت شرط جمله دلالت (p_i) و جملات KB (p'_i) یافت که آنها را یکسان سازی کند، آنگاه جمله حاصل از اعمال جایگزینی به متغیرهای قسمت نتیجه دلالت بدست می آید
- یک قانون بی عیب است

- نیاز به پیدا کردن جایگزینی هایی دارد که جملات منطقی مختلف را یکسان کند

استنتاج در منطق درجه اول

- یکسان سازی

- برای هر دو جمله در صورت امکان یک یکسان کننده (unifier) برمی گرداند

$$\text{UNIFY}(p, q) = \theta \text{ where } \text{SUBST}(\theta, p) = \text{SUBST}(\theta, q)$$

- اگر هر دو جمله از متغیر یکسانی استفاده کرده باشند قبل از اعمال یکسان سازی باید متغیرهای جملات استانداردسازی شوند: متغیرهای یکی از جملات باید تغییر

نام داده شوند $\text{UNIFY}(\text{Knows}(\text{John}, x), \text{Knows}(x, \text{Elizabeth})) = \text{fail}$

$$\text{UNIFY}(\text{Knows}(\text{John}, x), \text{Knows}(x_{17}, \text{Elizabeth})) = \{x/\text{Elizabeth}, x_{17}/\text{John}\}$$

- اگر بیش از یک یکسان کننده برای دو جمله وجود داشته باشد، پیدا کردن کلی ترین یکسان کننده (most general unifier – MGU) که یکتاست مطلوب تر است

- محدودیت های کمتری روی مقادیر متغیرها در نظر می گیرد

$$\begin{array}{ll} \text{UNIFY}(\text{Knows}(\text{John}, x), \text{Knows}(y, z)) & \{y/\text{John}, x/z\} \\ & \{y/\text{John}, x/\text{John}, z/\text{John}\} \end{array}$$

استنتاج در منطق درجه اول

- عبارتهای معین درجه اول

- جملات پایه

- جملات دلالت منطقی که قسمت شرط آنها ترکیب عطفی لفظهای مثبت و قسمت نتیجه یک لفظ مثبت تکی باشد

- جملات می‌توانند با استفاده از سور عمومی بیان شده باشند

- بعضی از لفظها متغیر هستند $King(x) \wedge Greedy(x) \Rightarrow Evil(x)$.

$King(John)$.

$Greedy(y)$.

- جملات

- اگر KB فقط شامل عبارتهای معین درجه اولی باشد که از هیچ نماد تابعی

استفاده نمی‌کنند به آن یک datalog می‌گویند

- استنتاج در اینگونه از KBها مزایایی دارد از جمله کمال

استنتاج در منطق درجه اول

- زنجیربندی جلورو
- مانند زنجیربندی جلورو برای منطق گزاره‌ای در KBهایی با عبارتهای شاخ قابل استفاده است
- با شروع از حقایق موجود، قانون شیوه تصدیق کلی را به جملاتی که شرط آنها برآورده شده باشد اعمال و نتیجه آنها را به KB اضافه می‌کند
- این فرآیند تا جایی تکرار می‌شود که یا جمله پرسش به KB اضافه شود یا هیچ جمله جدیدی اضافه نشود
- جمله‌ای جدید است که یک نامگذاری متفاوت (renaming) از یک جمله موجود در KB نباشد (تنها تفاوت در نام متغیرها نباشد)
- یک الگوریتم بی‌عیب و کامل است

استنتاج در منطق درجه اول

- مثال از زنجیربندی جلورو

The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.

- عبارات معین درجه اول بدست آمده

$$American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$$

$$\exists x Owns(Nono, x) \wedge Missile(x) \quad \longrightarrow \quad \begin{array}{l} Owns(Nono, M_1) \\ Missile(M_1) \end{array}$$

$$Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$$

$$Missile(x) \Rightarrow Weapon(x)$$

$$Enemy(x, America) \Rightarrow Hostile(x)$$

$$American(West)$$

$$Enemy(Nono, America)$$

استنتاج در منطق درجه اول

- مثال از زنجیربندی جلورو

$Owns(Nono, M_1)$

$Missile(M_1)$

$American(West)$

$Enemy(Nono, America)$

$American(West)$

$Missile(M_1)$

$Owns(Nono, M_1)$

$Enemy(Nono, America)$

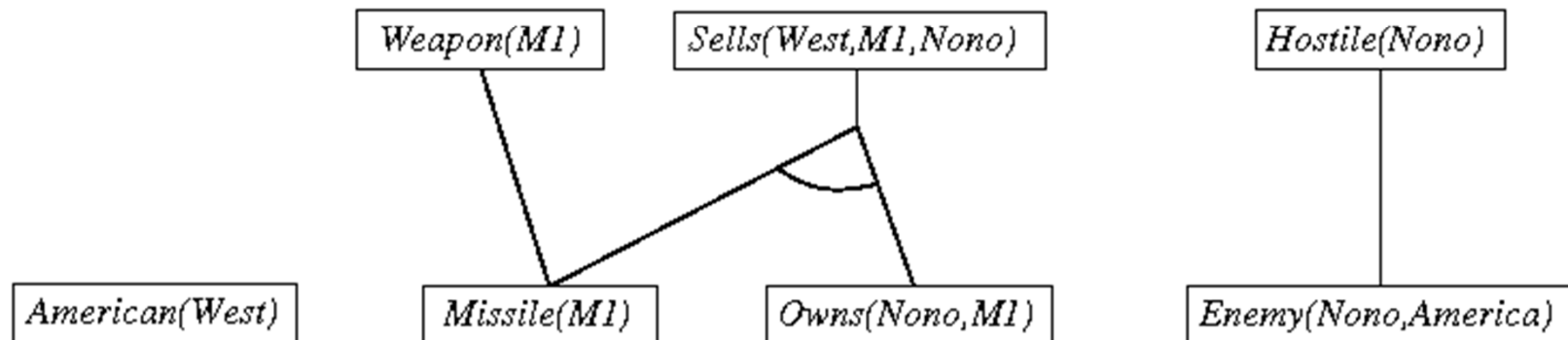
استنتاج در منطق درجه اول

- مثال از زنجیربندی جلورو

$Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$

$Missile(x) \Rightarrow Weapon(x)$

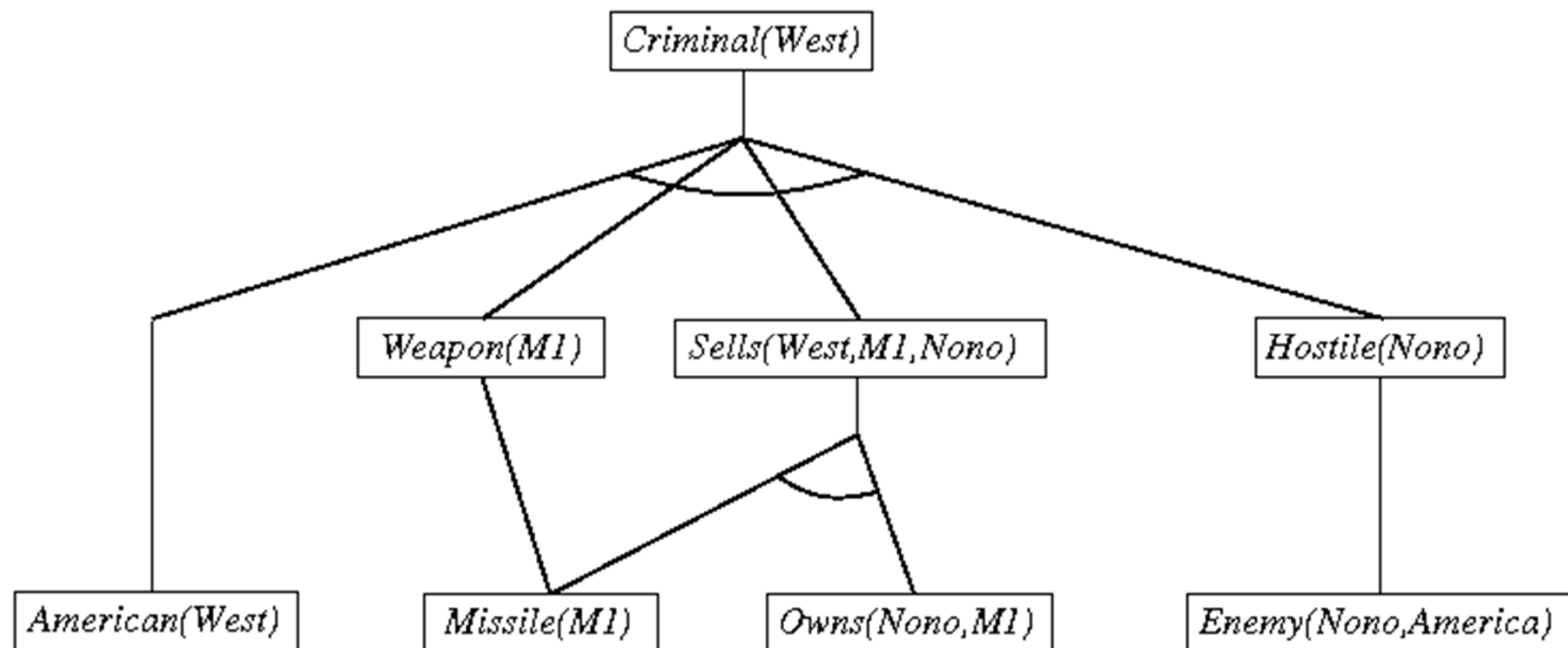
$Enemy(x, America) \Rightarrow Hostile(x)$



استنتاج در منطق درجه اول

- مثال از زنجیربندی جلورو

$$American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$$



استنتاج در منطق درجه اول

- زنجیربندی عقب‌رو
 - با شروع از جمله هدف (goal)، جملاتی که قسمت نتیجه آنها قابل یکسان‌سازی با جملات قبلی KB هستند را در نظر می‌گیرد و سپس برآورده شدن قسمت شرط این جملات را بررسی می‌کند
 - اگر جمله هدف دارای متغیر باشد، تمام جواب‌های ممکن پیدا می‌شوند
 - یک الگوریتم عمق اول (به صورت بازگشتی) است و دارای پیچیدگی فضایی خطی نسبت به تعداد جملات KB
 - در زبان برنامه نویسی منطق (Prolog) از این روش برای استنتاج استفاده می‌شود

استنتاج در منطق درجه اول

- مثال زنجیربندی عقبرو

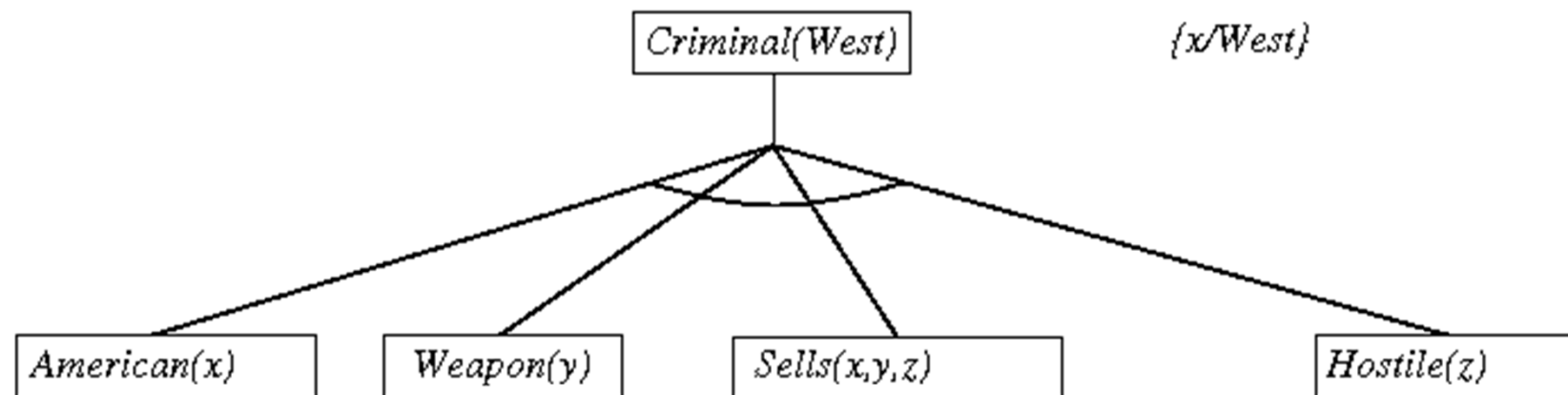
- جمله هدف: *Criminal(West)*

Criminal(West)

استنتاج در منطق درجه اول

- مثال زنجیربندی عقبرو

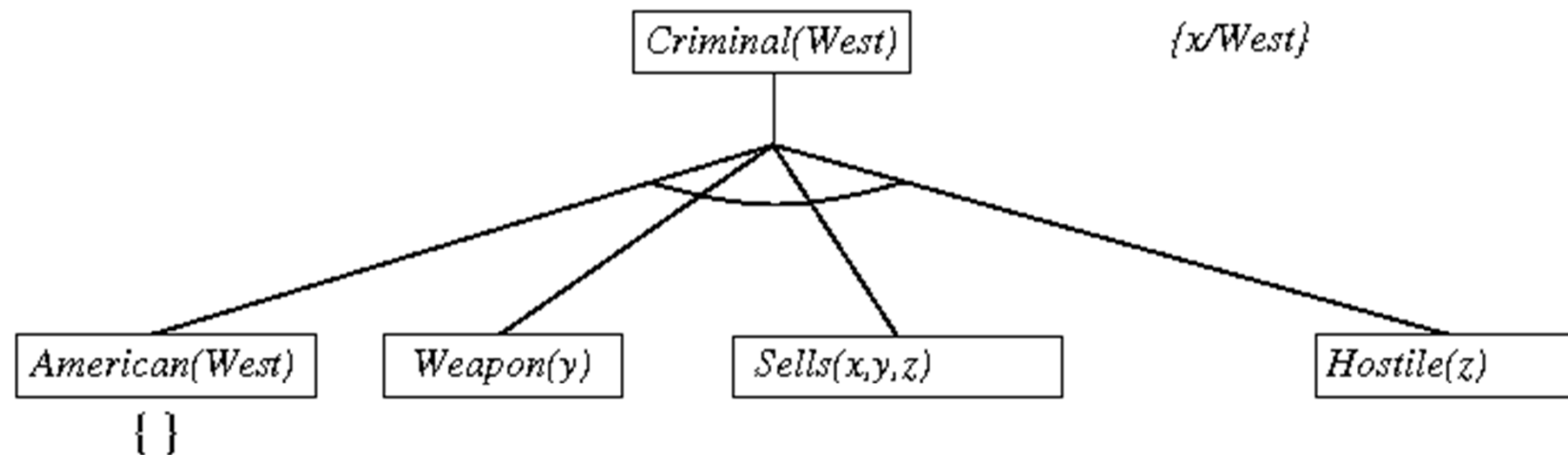
$$American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$$



استنتاج در منطق درجه اول

- مثال زنجیربندی عقبرو

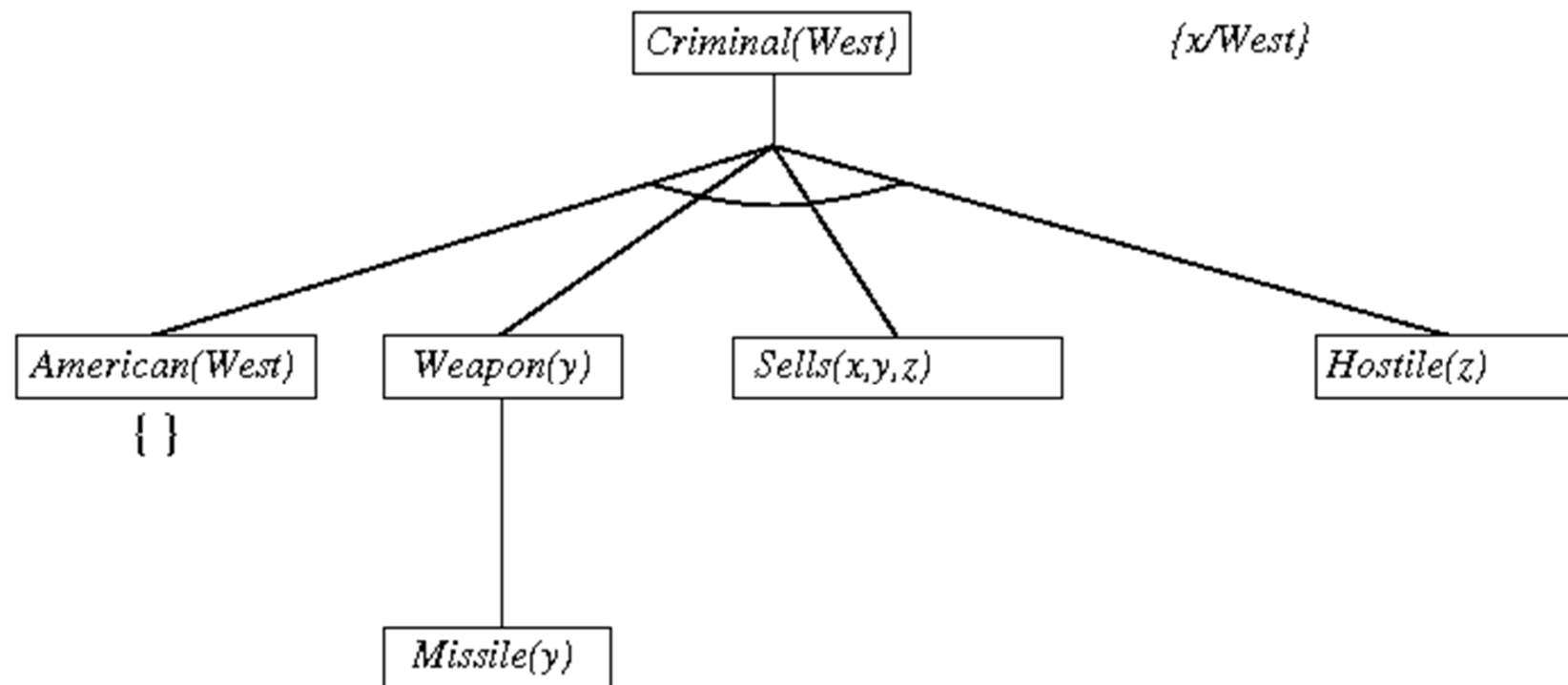
American(West)



استنتاج در منطق درجه اول

- مثال زنجیربندی عقبرو

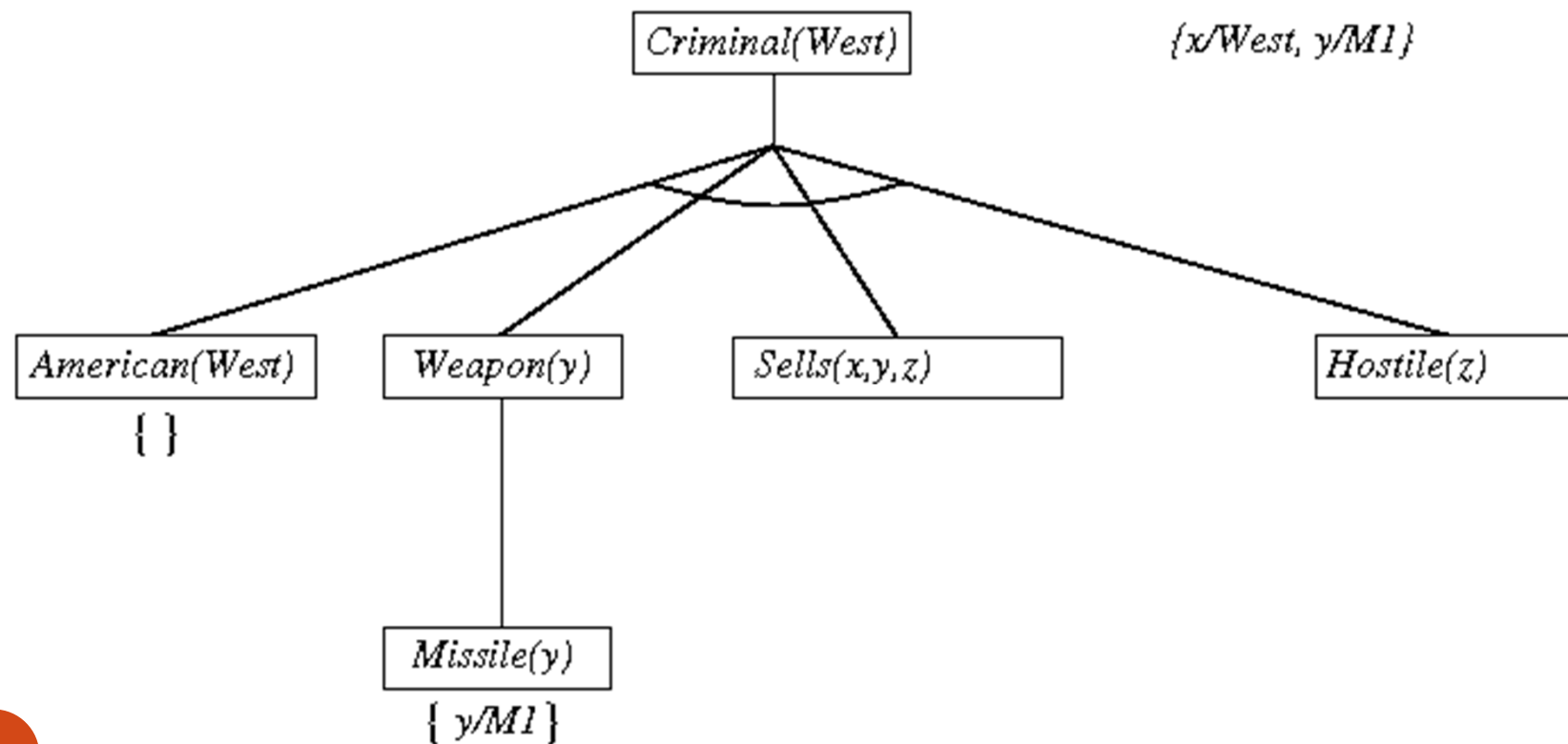
$$\text{Missile}(x) \Rightarrow \text{Weapon}(x)$$



استنتاج در منطق درجه اول

- مثال زنجیربندی عقبرو

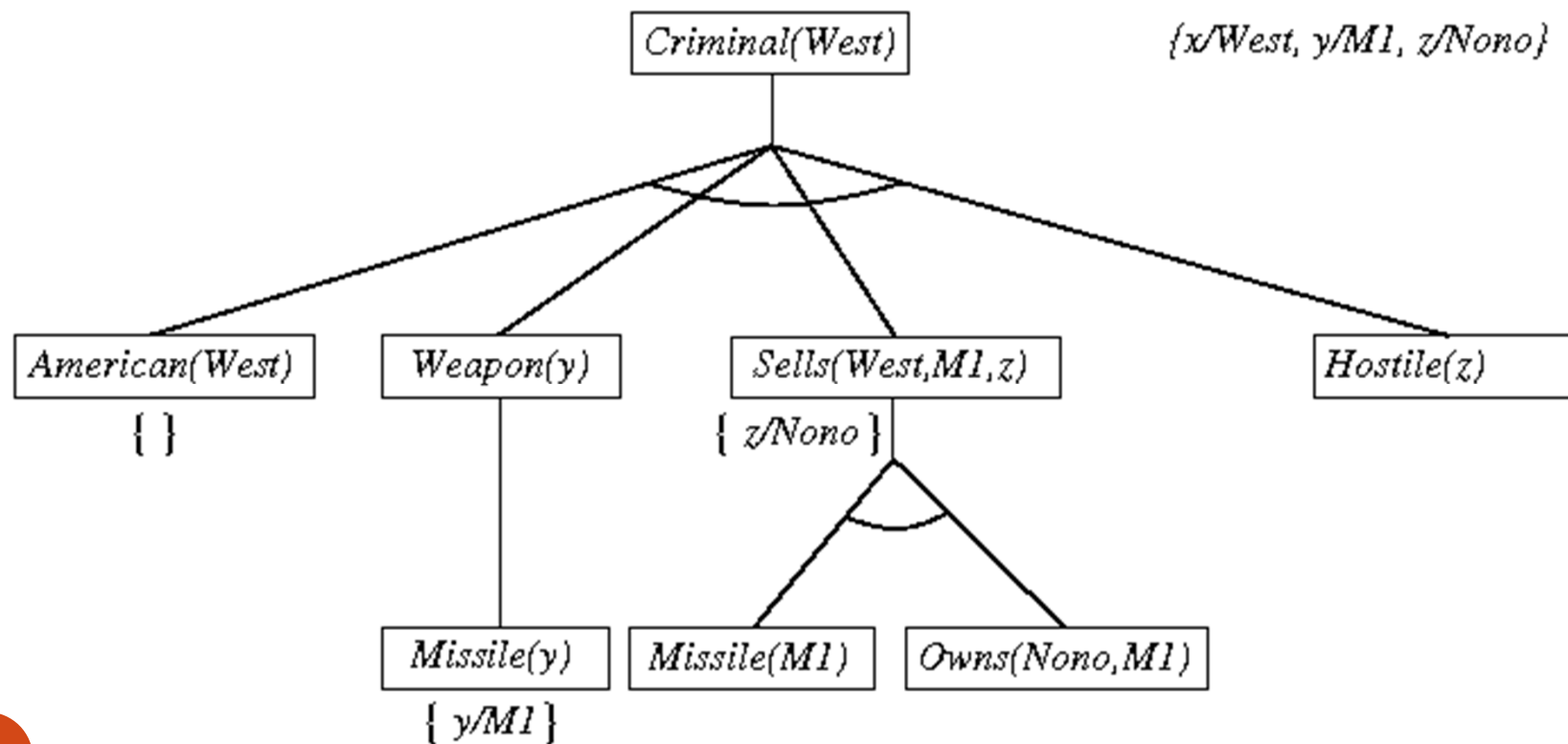
$Missile(M_1)$



استنتاج در منطق درجه اول

- مثال زنجیربندی عقب‌رو

$$\text{Missile}(x) \wedge \text{Owns}(\text{Nono}, x) \Rightarrow \text{Sells}(\text{West}, x, \text{Nono})$$



استنتاج در منطق درجه اول

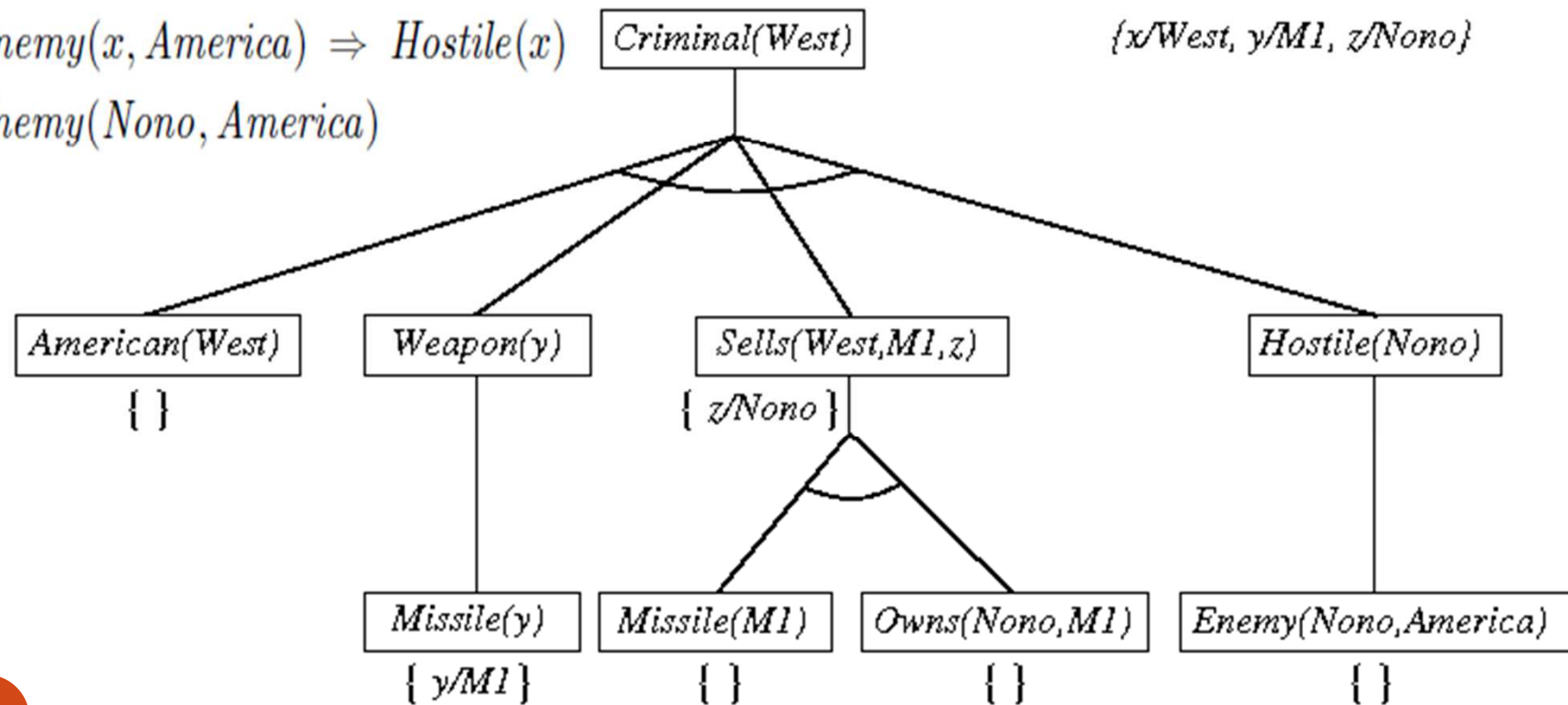
• مثال زنجیربندی عقب‌رو

$Owns(Nono, M_1)$

$Missile(M_1)$

$Enemy(x, America) \Rightarrow Hostile(x)$

$Enemy(Nono, America)$



استنتاج در منطق درجه اول

- مشکلات زنجیربندی عقب‌رو
- حالت‌های (اهداف میانی) تکراری
 - راه‌حل: بخاطر سپاری (memorization) اهداف میانی (subgoals) بدست آمده
 - حلقه‌های نامتناهی
 - راه‌حل: در نظر گرفتن یک پشته (stack) از اهداف برای جلوگیری از بررسی تودرتو
- قانون رفع (resolution) در FOL
 - استفاده از برهان خلف در اثبات جملات
 - جمله α از KB ملتزم می‌شود اگر $KB \wedge \neg \alpha$ ارضاء‌پذیر نباشد
 - جملات باید بصورت CNF باشند
 - هر جمله FOL می‌تواند به یک جمله CNF که معادل استنتاجی (inferentially equivalent) آن است تبدیل شود

استنتاج در منطق درجه اول

• تبدیل جملات FOL به CNF

$$\forall x [\forall y \text{ Animal}(y) \Rightarrow \text{Loves}(x, y)] \Rightarrow [\exists y \text{ Loves}(y, x)]$$

1. حذف دلالت‌ها

$$\forall x [\neg \forall y \neg \text{Animal}(y) \vee \text{Loves}(x, y)] \vee [\exists y \text{ Loves}(y, x)]$$

2. قراردادن حروف نفی فقط در پشت واژه‌های بنیادی

$$\forall x [\exists y \neg(\neg \text{Animal}(y) \vee \text{Loves}(x, y))] \vee [\exists y \text{ Loves}(y, x)]$$

$$\forall x [\exists y \neg \neg \text{Animal}(y) \wedge \neg \text{Loves}(x, y)] \vee [\exists y \text{ Loves}(y, x)]$$

$$\forall x [\exists y \text{ Animal}(y) \wedge \neg \text{Loves}(x, y)] \vee [\exists y \text{ Loves}(y, x)]$$

3. استاندارد کردن متغیرها

$$\forall x [\exists y \text{ Animal}(y) \wedge \neg \text{Loves}(x, y)] \vee [\exists z \text{ Loves}(z, x)]$$

استنتاج در منطق درجه اول

• تبدیل جملات FOL به CNF (ادامه)

4. Skolem کردن (Skolemization) سورهای وجودی

- اگر سور وجودی در یک سور عمومی باشد بجای نمونه گذاری وجودی (EI) از توابع Skolem استفاده می شود

$$\forall x [Animal(F(x)) \wedge \neg Loves(x, F(x))] \vee Loves(G(x), x)$$

5. حذف سورهای عمومی

$$[Animal(F(x)) \wedge \neg Loves(x, F(x))] \vee Loves(G(x), x)$$

6. توزیع ترکیب فصلی روی ترکیب عطفی

$$[Animal(F(x)) \vee Loves(G(x), x)] \wedge [\neg Loves(x, F(x)) \vee Loves(G(x), x)]$$

استنتاج در منطق درجه اول

- قانون رفع

- نسخه ترفیع یافته قانون رفع در منطق گزاره‌ای و دارای دو مرحله
- رفع دوتایی (binary): اگر در دو عبارت، دو لفظ وجود داشته که بتوان یک از آنها را با نفی دیگری یکسان سازی کرد، $\text{UNIFY}(\ell_i, \neg m_j) = \theta$ ، آنگاه

$$\frac{\ell_1 \vee \dots \vee \ell_k, \quad m_1 \vee \dots \vee m_n}{\text{SUBST}(\theta, \ell_1 \vee \dots \vee \ell_{i-1} \vee \ell_{i+1} \vee \dots \vee \ell_k \vee m_1 \vee \dots \vee m_{j-1} \vee m_{j+1} \vee \dots \vee m_n)}$$

- ساده سازی: لفظهای قابل یکسان سازی (یکسان سازی در نظر گرفته شده در هنگام رفع دوتایی) باید در عبارت حاصل از رفع به یک لفظ تقلیل داده شوند
- قانون رفع (ترکیب رفع دوتایی و ساده سازی) کامل است

استنتاج در منطق درجه اول

• مثال ۱: اعمال قانون رفع

• KB بصورت CNF:

$\neg American(x) \vee \neg Weapon(y) \vee \neg Sells(x, y, z) \vee \neg Hostile(z) \vee Criminal(x)$

$\neg Missile(x) \vee \neg Owns(Nono, x) \vee Sells(West, x, Nono)$

$\neg Enemy(x, America) \vee Hostile(x)$

$\neg Missile(x) \vee Weapon(x)$

$Owns(Nono, M_1)$

$American(West)$

$Missile(M_1)$

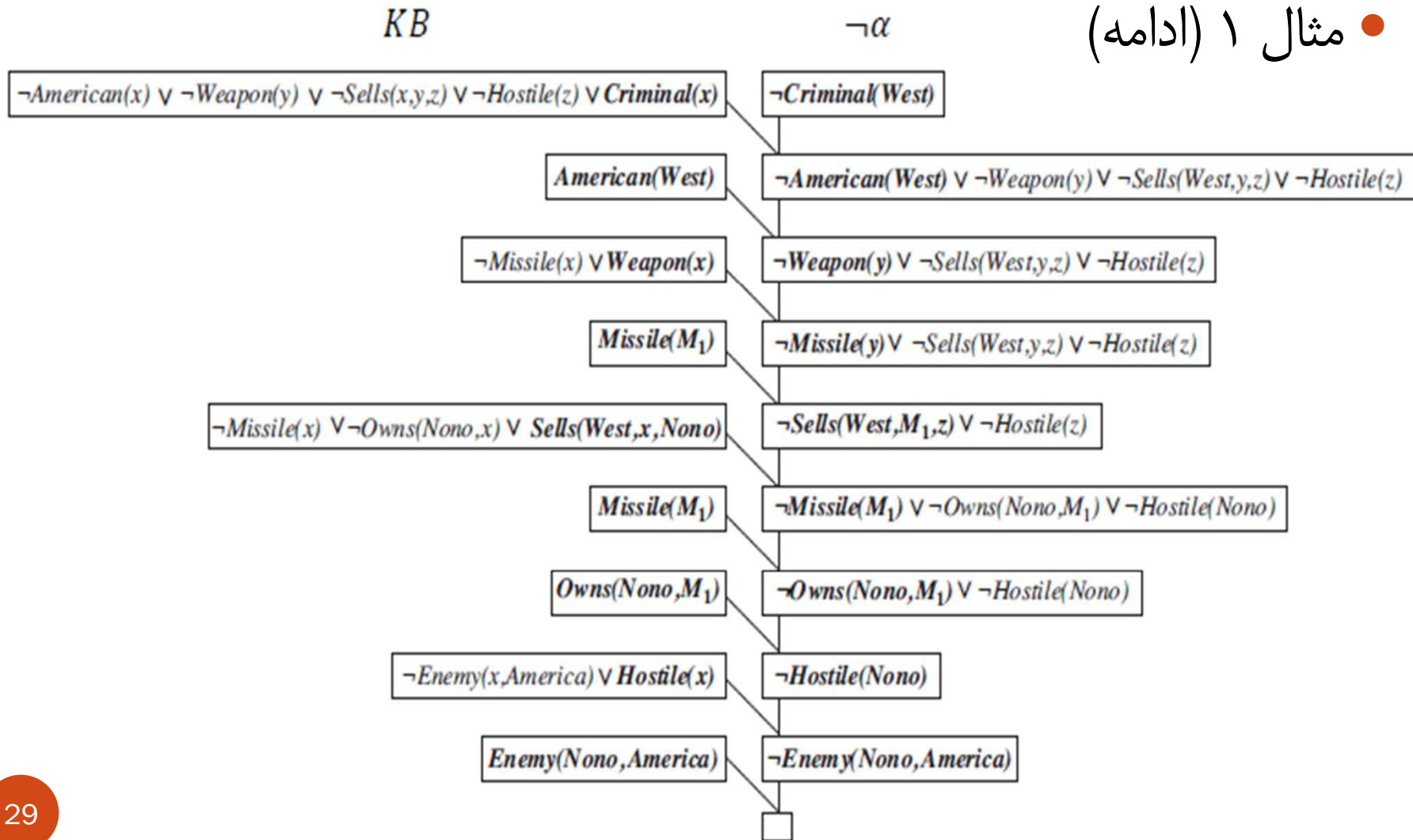
$Enemy(Nono, America)$.

$\neg Criminal(West)$

• هدف نفی شده:

استنتاج در منطق درجه اول

• مثال ۱ (ادامه)



استنتاج در منطق درجه اول

• مثال ۲: اعمال قانون رفع

Everyone who loves all animals is loved by someone.

Anyone who kills an animal is loved by no one.

Jack loves all animals.

Either Jack or Curiosity killed the cat, who is named Tuna.

Did Curiosity kill the cat?

• بیان مسأله با زبان FOL

$$\forall x [\forall y \text{ Animal}(y) \Rightarrow \text{Loves}(x, y)] \Rightarrow [\exists y \text{ Loves}(y, x)]$$

$$\forall x [\exists z \text{ Animal}(z) \wedge \text{Kills}(x, z)] \Rightarrow [\forall y \neg \text{Loves}(y, x)]$$

$$\forall x \text{ Animal}(x) \Rightarrow \text{Loves}(\text{Jack}, x)$$

$$\text{Kills}(\text{Jack}, \text{Tuna}) \vee \text{Kills}(\text{Curiosity}, \text{Tuna})$$

$$\text{Cat}(\text{Tuna})$$

$$\forall x \text{ Cat}(x) \Rightarrow \text{Animal}(x)$$

$$\neg \text{Kills}(\text{Curiosity}, \text{Tuna})$$

• هدف نفی شده:

استنتاج در منطق درجه اول

• مثال ۲ (ادامه)

• KB بصورت CNF

$Animal(F(x)) \vee Loves(G(x), x)$

$\neg Loves(x, F(x)) \vee Loves(G(x), x)$

$\neg Loves(y, x) \vee \neg Animal(z) \vee \neg Kills(x, z)$

$\neg Animal(x) \vee Loves(Jack, x)$

$Kills(Jack, Tuna) \vee Kills(Curiosity, Tuna)$

$Cat(Tuna)$

$\neg Cat(x) \vee Animal(x)$

$\neg Kills(Curiosity, Tuna)$

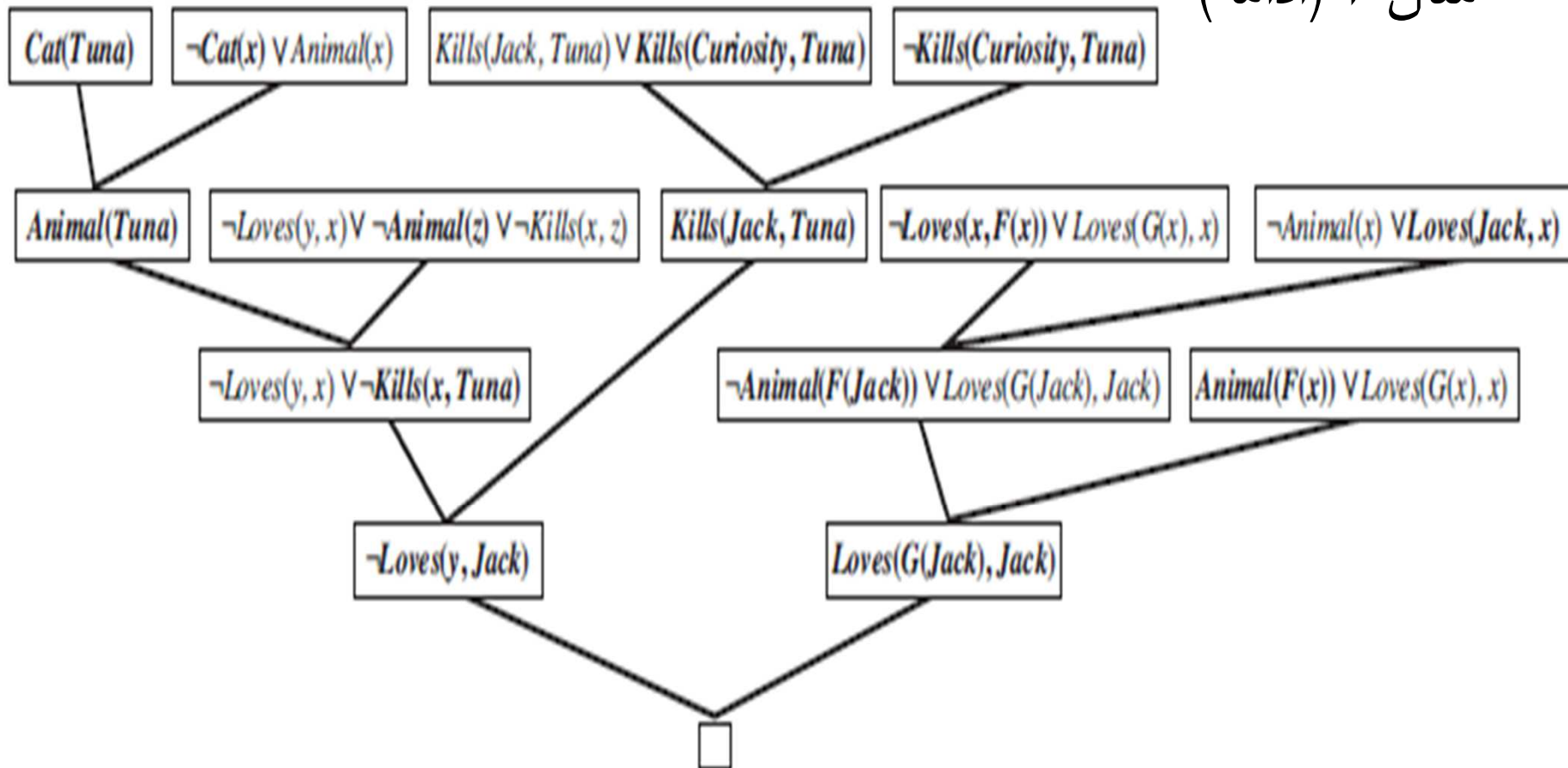
• هدف نفی شده

• هدف می تواند $\exists w Kills(w, Tuna)$ باشد: چه کسی Tuna را کشته است

• نفی آن به صورت CNF: $\neg Kills(w, Tuna)$

استنتاج در منطق درجه اول

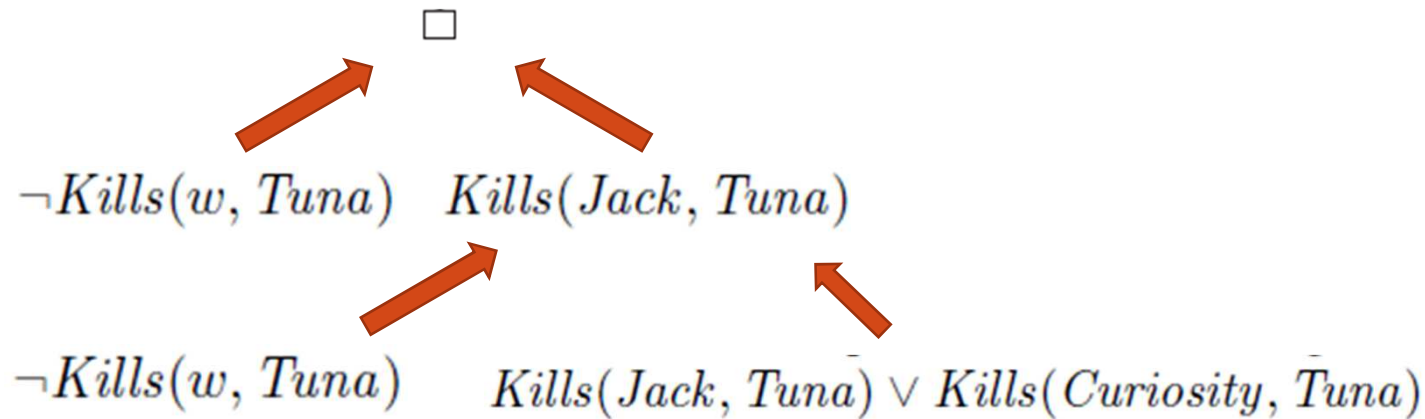
• مثال ۲ (ادامه)



استنتاج در منطق درجه اول

- مقداردهی‌های متفاوت به یک متغیر در یک مسیر از اثبات می‌تواند منجر به اثبات‌های غیرسازنده (non-constructive proofs) شود

• مثال



• راهکار

- در هر اثبات هر متغیر فقط یکبار مقداردهی شود
- استفاده از یک مسند پاسخ در کنار هدف نفی‌شده: $\neg Kills(w, Tuna) \vee Answer(w)$

استنتاج در منطق درجه اول

- الگوریتم مبتنی بر قانون رفع کامل است
- اگر اثباتی برای جمله هدف وجود داشته باشد آن را پیدا می کند
- راهبردهای اعمال کارآمدتر قانون رفع
 - ترجیح تکی (unit preference): یکی از جملات عبارت تکی (یک لفظ) باشد
 - رفع تکی روی KBهایی با عبارات شاخ کامل است (مانند زنجیربندی جلورو عمل می کند)
 - مجموعه پشتیبان (set of support): اولویت دهی به جملات حاوی یکی از عبارات در نظر گرفته شده (مجموعه پشتیبان)
 - رفع ورودی: یکی از جملات رفع همیشه ورودی (اصول اولیه یا پرسش هدف) باشد
 - نتیجه گیری (subsumption): جملات قابل استنتاج از سایر جملات را از KB حذف می کند: برای کوچک نگه داشتن KB

استنتاج در منطق درجه اول

- استنتاج برای رابطه برابری (=)

- تعریف اصول ناظر به برابری در KB: بازتابی، تقارن، تعدی، ...

- تعداد اصول لازم زیاد است

- استفاده از قوانین استنتاج خاص

- قوانین demodulation و paramodulation: یکسان سازی و جایگزینی لفظهای

- معادل با یکدیگر

$$P(F(x, B), x) \vee Q(x)$$

$$F(A, y) = y \vee R(y)$$

$$P(B, A) \vee Q(A) \vee R(B) \quad \theta = \text{UNIFY}(F(A, y), F(x, B)) = \{x/A, y/B\}$$

- گسترش الگوریتم یکسان سازی برای در نظر گرفتن برابری

- مثال: یکسان سازی $1 + 2$ و $2 + 1$ با در نظر گرفتن $x + y = y + x$

برنامه‌نویسی منطق (Prolog)

• بر اساس ایده $\text{Algorithm} = \text{Logic} + \text{Control}$

• معادله Robert Kowalski

• برای کار با نمادها

• در تولید سریع نمونه کارها (prototype)

• در نوشتن کامپایلرها و پارسرهای زبان‌های طبیعی

• تولید سیستم‌های خبره

• برنامه‌ها نشان‌دهنده‌ی KB هستند

• جملات KB بصورت عبارات معین هستند

برنامه نویسی منطق

• ساختار جملات زبان

- متغیرها با حروف بزرگ شروع می شوند
 - ثابت‌ها با حروف کوچک شروع می شوند
 - ترکیب عطفی با ویرگول (,) نشان داده می شود
 - ترکیب فصلی با «نقطه – ویرگول» (;) نشان داده می شود
 - قسمت نتیجه و شرط عبارت‌ها با “:-” از هم جدا می شوند
- $$A \wedge B \Rightarrow C \rightarrow C :- A, B$$
- جملات پرسشی به صورت “A-?” پرسیده می شوند

برنامه‌نویسی منطق

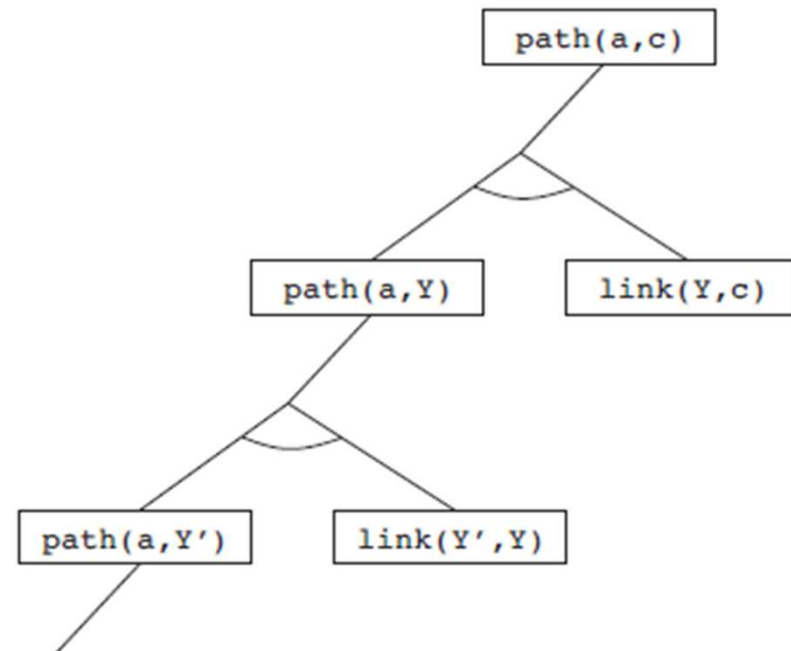
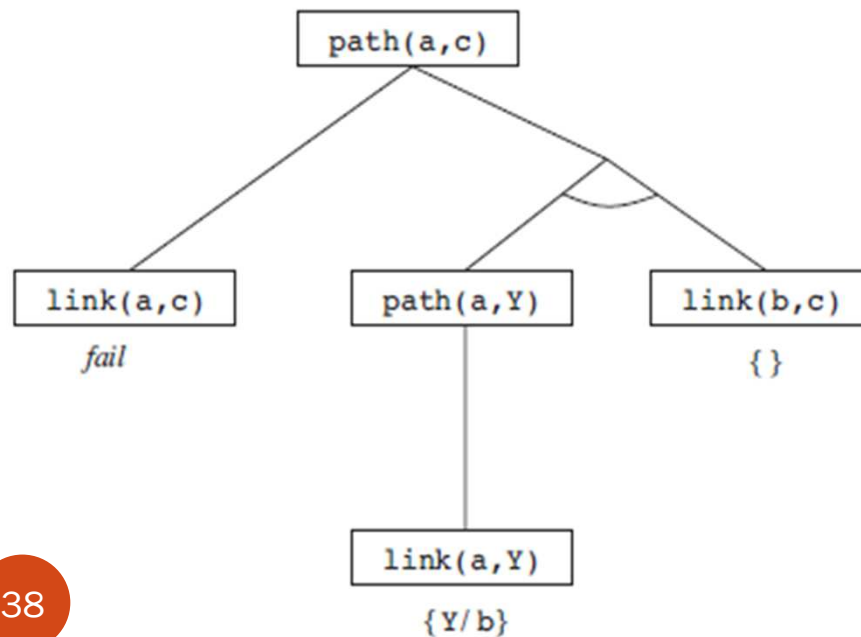
- اجرای برنامه‌ها (استنتاج)
- از روش زنجیربندی عقب‌رو استفاده می‌شود
- ترتیب جملات در اولویت بررسی آنها نقش دارد

```
path(X,Z) :- link(X,Z).
```

```
path(X,Z) :- path(X,Y), link(Y,Z).
```

```
path(X,Z) :- path(X,Y), link(Y,Z).
```

```
path(X,Z) :- link(X,Z).
```



جمع‌بندی

- رویکردهای مختلف برای استنتاج در منطق درجه اول
- گزاره‌ای‌سازی و استفاده از استنتاج در منطق گزاره‌ای
- یکسان‌سازی و ترفیع قوانین استنتاج برای منطق درجه اول
- الگوریتم‌های زنجیره‌بندی جلورو و عقب‌رو
- استنتاج با کمک قانون رفع کلی
- برنامه‌نویسی منطق