

SZTUCZNA INTELIGENCJA I SYSTEMY DORADCZE

WNIOSKOWANIE W LOGICE I RZĘDU

Rachunek zdań: zalety i wady

- ☺ Rachunek zdań jest *deklaratywny*:
elementy syntaktyki odpowiadają faktom
- ☺ Rach. zdań dopuszcza częściową/alternatywną/zanegowaną informację
(w przeciwieństwie do większości struktur danych i baz danych)
- ☺ Rachunek zdań jest *składnikowy*:
znaczenie $B_{1,1} \wedge P_{1,2}$ wynika ze znaczenia $B_{1,1}$ i $P_{1,2}$
- ☺ Znaczenie w rachunku zdań jest *niezależne od kontekstu*
(w przeciwieństwie do języka naturalnego)
- ☹ Rachunek zdań ma bardzo ograniczoną moc wyrażania
(w przeciwieństwie do języka naturalnego), np.
nie da się wyrazić zdania “pułapki powodują wiatr w sąsiednich polach”
inaczej niż przez napisanie oddzielnego zdania dla każdego pola

Logika I rzędu

Rachunek zdań zakłada, że świat zawiera tylko *fakty*, natomiast logika I rzędu (bliższa językowi naturalnemu) zakłada, że świat zawiera:

- **Obiekty**: ludzie, domy, liczby, teorie, Ronald McDonald, kolory, baseball, wojny, wieki ...
- **Relacje**: czerwony, okrągły, pierwszy, wieloodcinkowy ..., brat, większy niż, w środku, część, ma kolor, zdarzyło się po, posiada, pomiędzy, ...
- **Funkcje**: ojciec, najlepszy przyjaciel, trzeci właściciel, o jeden więcej niż, początek ...

Syntaktyka

Stałe	<i>KingJohn, 2, UCB, ...</i>
Predykaty	<i>Brother, >, ...</i>
Funkcje	<i>Sqrt, LeftLegOf, ...</i>
Zmienne	<i>x, y, a, b, ...</i>
Łączniki	$\wedge \vee \neg \Rightarrow \Leftrightarrow$
Równość	$=$
Kwantyfikatory	$\forall \exists$

Syntaktyka: zdania atomowe

Zdanie atomowe = $predicate(term_1, \dots, term_n)$
lub $term_1 = term_2$

Term = $function(term_1, \dots, term_n)$
lub *constant* or *variable*

Np. $Brother(KingJohn, RichardTheLionheart)$
> $(Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))$

Syntaktyka: zdania złożone

Zdania złożone zbudowane są ze zdań atomowych przy pomocy łączników

$$\neg S, \quad S_1 \wedge S_2, \quad S_1 \vee S_2, \quad S_1 \Rightarrow S_2, \quad S_1 \Leftrightarrow S_2$$

Np. $Sibling(KingJohn, Richard) \Rightarrow Sibling(Richard, KingJohn)$

$$>(1, 2) \vee \leq(1, 2)$$

$$>(1, 2) \wedge \neg >(1, 2)$$

Semantyka

Zdania są prawdziwe względem modelu i interpretacji

Model zawiera ≥ 1 obiektów (elementów dziedziny) i relacje między nimi

Interpretacja specyfikuje przyporządkowania:

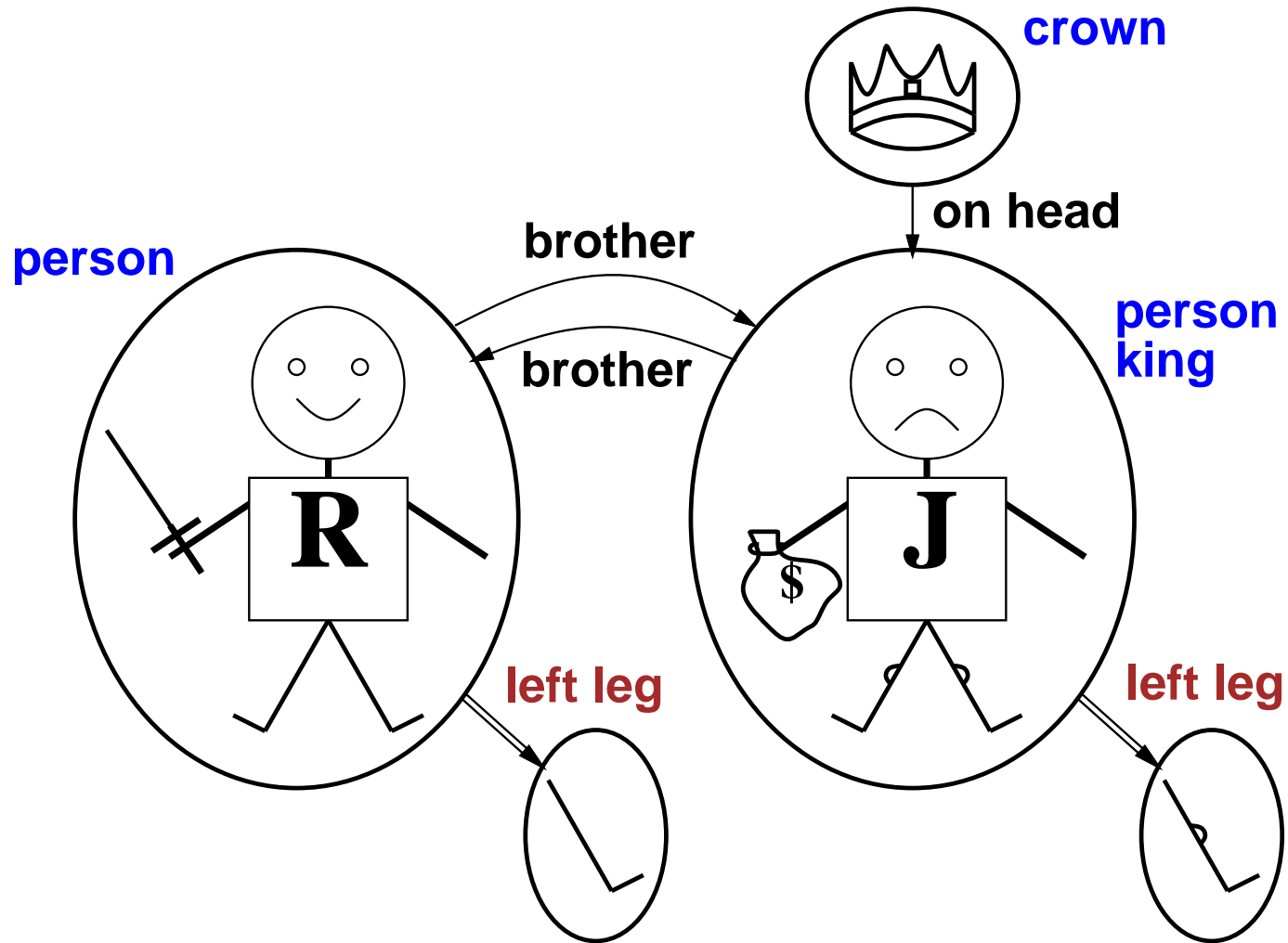
symbole stałe \rightarrow obiekty

predykaty \rightarrow relacje

symbole funkcyjne \rightarrow relacje funkcyjne

Zdanie atomowe $predicate(term_1, \dots, term_n)$ jest prawdziwe wtw obiekty przyporządkowane do $term_1, \dots, term_n$ są w relacji przyporządkowanej do $predicate$

Model: przykład



Modele logiki I rzędu: bardzo dużo!

Można wyliczać modele dla danego słownika bazy wiedzy KB :

Dla każdej liczby elementów dziedziny n od 1 do ∞

 Dla każdego k -arnego predykatu P_k w słowniku

 Dla każdej możliwej relacji k -arnej na n obiektach

 Dla każdego symbolu stałego C w słowniku

 Dla każdego przyporządkowania C do jednego z n obiektów ...

Określanie logicznych konsekwencji przez wyliczanie modeli jest niepraktyczne!

Kwantyfikatory uniwersalne

$\forall \langle variables \rangle \langle sentence \rangle$

Każdy w Berkeley jest sprytny:

$\forall x \text{ At}(x, Berkeley) \Rightarrow \text{Smart}(x)$

$\forall x P$ jest prawdziwe w modelu m wtw P jest prawdziwe z x przyporządkowanym do *każdego* możliwego obiektu w modelu

Nieformalnie, równoważne **koniunkcji** wszystkich **podstawień** w P

$\text{At}(KingJohn, Berkeley) \Rightarrow \text{Smart}(KingJohn)$
 $\wedge \text{At}(Richard, Berkeley) \Rightarrow \text{Smart}(Richard)$
 $\wedge \text{At}(Berkeley, Berkeley) \Rightarrow \text{Smart}(Berkeley)$
 $\wedge \dots$

Kwantyfikatory egzystencjalne

$\exists \langle variables \rangle \langle sentence \rangle$

Ktoś w Stanford jest sprytny:

$\exists x \text{ At}(x, \text{Stanford}) \wedge \text{Smart}(x)$

$\exists x P$ jest prawdziwe w modelu m wtw P jest prawdziwe z x przyporządkowanym do pewnego możliwego obiektu w modelu

Nieformalnie, równoważne **alternatywie podstawień** w P

- $\text{At}(\text{KingJohn}, \text{Stanford}) \wedge \text{Smart}(\text{KingJohn})$
- $\vee \text{At}(\text{Richard}, \text{Stanford}) \wedge \text{Smart}(\text{Richard})$
- $\vee \text{At}(\text{Stanford}, \text{Stanford}) \wedge \text{Smart}(\text{Stanford})$
- $\vee \dots$

Kwantyfikatory: własności

$\forall x \forall y$ równoważne $\forall y \forall x$

$\exists x \exists y$ równoważne $\exists y \exists x$

$\exists x \forall y$ **nie** jest równoważne $\forall y \exists x$

$\exists x \forall y \text{ Loves}(x, y)$

“Istnieje osoba, która kocha wszystkich”

$\forall y \exists x \text{ Loves}(x, y)$

“Każdy jest kochany przez co najmniej jedną osobę”

Dualność kwantyfikatorów: jeden można opisać przy pomocy drugiego

$\forall x \text{ Likes}(x, \text{IceCream}) \quad \neg \exists x \neg \text{Likes}(x, \text{IceCream})$

$\exists x \text{ Likes}(x, \text{Broccoli}) \quad \neg \forall x \neg \text{Likes}(x, \text{Broccoli})$

Tłumaczenie języka naturalnego do I rzędu

Bracia są rodzeństwem

Tłumaczenie języka naturalnego do I rzędu

Bracia są rodzeństwem

$\forall x, y \text{ Brother}(x, y) \Rightarrow \text{Sibling}(x, y).$

“Rodzeństwo” jest symetryczne

Tłumaczenie języka naturalnego do I rzędu

Bracia są rodzeństwem

$$\forall x, y \text{ Brother}(x, y) \Rightarrow \text{Sibling}(x, y).$$

“Rodzeństwo” jest symetryczne

$$\forall x, y \text{ Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x).$$

Czyjaś matka jest kobietą i jego rodzicem

Tłumaczenie języka naturalnego do I rzędu

Bracia są rodzeństwem

$$\forall x, y \text{ Brother}(x, y) \Rightarrow \text{Sibling}(x, y).$$

“Rodzeństwo” jest symetryczne

$$\forall x, y \text{ Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x).$$

Czyjaś matka jest kobietą i jego rodzicem

$$\forall x, y \text{ Mother}(x, y) \Leftrightarrow (\text{Female}(x) \wedge \text{Parent}(x, y)).$$

Kuzyn jest dzieckiem rodzeństwa rodzica

Tłumaczenie języka naturalnego do I rzędu

Bracia są rodzeństwem

$$\forall x, y \text{ Brother}(x, y) \Rightarrow \text{Sibling}(x, y).$$

“Rodzeństwo” jest symetryczne

$$\forall x, y \text{ Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x).$$

Czyjaś matka jest kobietą i jego rodzicem

$$\forall x, y \text{ Mother}(x, y) \Leftrightarrow (\text{Female}(x) \wedge \text{Parent}(x, y)).$$

Kuzyn jest dzieckiem rodzeństwa rodzica

$$\forall x, y \text{ FirstCousin}(x, y) \Leftrightarrow \exists p, ps \text{ Parent}(p, x) \wedge \text{Sibling}(ps, p) \wedge \text{Parent}(ps, y)$$

Rownosc

$term_1 = term_2$ jest prawdziwe w danej interpretacji wtedy i tylko wtedy jeśli $term_1$ i $term_2$ przyporządkowane są do tego samego obiektu

Np. $1 = 2$ i $\forall x \times(Sqrt(x), Sqrt(x)) = x$ są spełnialne
 $2 = 2$ jest tautologią

Np. definicja (pełnego) *Sibling* w terminach *Parent*:

$$\forall x, y \text{ Sibling}(x, y) \Leftrightarrow [\neg(x = y) \wedge \exists m, f \neg(m = f) \wedge \\ \text{Parent}(m, x) \wedge \text{Parent}(f, x) \wedge \text{Parent}(m, y) \wedge \text{Parent}(f, y)]$$

Baza wiedzy w języku I rzędu

Agent w świecie Wumpusa

odczuwający zapach i wiatr, ale nie obserwujący błysku w chwili $t = 5$:

$Tell(KB, Percept([Smell, Breeze, None], 5))$

$Ask(KB, \exists a Action(a, 5))$

tzn. czy z KB wynikają jakieś konkretne akcje w chwili $t = 5$?

Odpowiedź: *Tak*, $\{a/Shoot\}$ ← podstawienie (lista powiązań)

Dla danego zdania S i podstawienia σ ,

$S\sigma$ oznacza wynik zastosowania σ do S ; np.

$S = Smarter(x, y)$

$\sigma = \{x/Hillary, y/Bill\}$

$S\sigma = Smarter(Hillary, Bill)$

$Ask(KB, S)$ zwraca pewne/wszystkie σ takie, że $KB \models S\sigma$

Baza wiedzy: przykład

“Obserwacja”

$$\forall b, g, t \text{ Percept}([Smell, b, g], t) \Rightarrow Smelt(t)$$
$$\forall s, b, t \text{ Percept}([s, b, Glitter], t) \Rightarrow AtGold(t)$$

Reakcja: $\forall t \text{ AtGold}(t) \Rightarrow \text{Action}(Grab, t)$

Reakcja z wewnętrznym stanem: czy nie mamy już złota?

$$\forall t \text{ AtGold}(t) \wedge \neg \text{Holding}(Gold, t) \Rightarrow \text{Action}(Grab, t)$$

Baza wiedzy: przykład

Własności miejsca:

$$\forall x, t \text{ At}(\text{Agent}, x, t) \wedge \text{Smelt}(t) \Rightarrow \text{Smelly}(x)$$

$$\forall x, t \text{ At}(\text{Agent}, x, t) \wedge \text{Breeze}(t) \Rightarrow \text{Breezy}(x)$$

Blisko pułapek jest wiatr:

Reguła *diagnostyczna* — wnioskuje przyczynę z efektu

$$\forall y \text{ Breezy}(y) \Rightarrow \exists x \text{ Pit}(x) \wedge \text{Adjacent}(x, y)$$

Reguła *przyczynowa* — wnioskuje efekt z przyczyny

$$\forall x, y \text{ Pit}(x) \wedge \text{Adjacent}(x, y) \Rightarrow \text{Breezy}(y)$$

Definicja: $\forall y \text{ Breezy}(y) \Leftrightarrow [\exists x \text{ Pit}(x) \wedge \text{Adjacent}(x, y)]$

Baza wiedzy: przykład

Własności miejsca:

$$\forall x, t \text{ At}(\text{Agent}, x, t) \wedge \text{Smelt}(t) \Rightarrow \text{Smelly}(x)$$

$$\forall x, t \text{ At}(\text{Agent}, x, t) \wedge \text{Breeze}(t) \Rightarrow \text{Breezy}(x)$$

Blisko pułapek jest wiatr:

Reguła **diagnostyczna** — wnioskuje przyczynę z efektu

$$\forall y \text{ Breezy}(y) \Rightarrow \exists x \text{ Pit}(x) \wedge \text{Adjacent}(x, y)$$

Reguła **przyczynowa** — wnioskuje efekt z przyczyny

$$\forall x, y \text{ Pit}(x) \wedge \text{Adjacent}(x, y) \Rightarrow \text{Breezy}(y)$$

Definicja: $\forall y \text{ Breezy}(y) \Leftrightarrow [\exists x \text{ Pit}(x) \wedge \text{Adjacent}(x, y)]$

OBSERWACJA: logika I rzędu ma dużo większą moc wyrażania niż rachunek zdań

Wnioskowanie w logice I rzędu

- ◇ Redukcja do wnioskowania w rachunku zdań
- ◇ Unifikacja + uogólniony Modus Ponens
 - Forward chaining
 - Backward chaining (Prolog)
- ◇ Rezolucja

Wnioskowanie w logice: historia

450B.C.	Stoicy	rachunek zdań, wnioskowanie (prawdopodobnie)
322B.C.	Arystoteles	“sylogizmy” (reguły wnioskowanie), kwantyfikatory
1565	Cardano	teoria prawd. (rachunek zdań + niepewność)
1847	Boole	rachunek zdań (ponownie)
1879	Frege	logika I rzędu
1922	Wittgenstein	dowodzenie przez tabele prawdziwości
1930	Gödel	\exists pełny algorytm dla log. I rzędu
1930	Herbrand	pełny algorytm dla log. I rzędu (redukcja do rach. zdań)
1931	Gödel	$\neg\exists$ pełny algorytm dla arytmetyki
1960	Davis/Putnam	“praktyczny” algorytm dla rachunku zdań
1965	Robinson	“praktyczny” algorytm dla log. I rzędu — rezolucja

Instancjacja uniwersalna (UI)

Każda instancjacja uniwersalnie kwantyfikowanego zdania jest logiczną konsekwencją reguły:

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

dla dowolnej zmiennej v i termu ustalonego g

Np. $\forall x \text{ King}(x) \wedge \text{Greedy}(x) \Rightarrow \text{Evil}(x)$ daje

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

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

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

⋮

Instancjacja egzystencjalna (EI)

Dla każdego zdania α , zmiennej v , i symbolu stałego k
który nie występuje nigdzie w bazie wiedzy:

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

Np. $\exists x \text{Crown}(x) \wedge \text{OnHead}(x, \text{John})$ pociąga

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

przy założeniu, że C_1 jest nowym symbolem stałym, nazywanym **stałą Skolema**

Inny przykład: z $\exists x \text{d}(x^y)/dy = x^y$ otrzymujemy

$$\text{d}(e^y)/dy = e^y$$

przy założeniu, że e jest nowym symbolem stałym

Instancjacja: własności

Instancjacja uniwersalna może być stosowana kilkakrotnie,
żeby *dodać* nowe zdania;
nowa KB jest logicznie równoważna poprzedniej

Instancjacja egzystencjalna może być zastosowana raz,
żeby *zamienić* zdania z kwantyfikatorami egzystencjalnymi;
nowa KB *nie* jest równoważna poprzedniej,
ale jest spełnialna wtw kiedy poprzednia KB jest spełnialna

Redukcja do rachunku zdań

Założmy, że mamy daną następującą bazę wiedzy KB:

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

$$\text{King}(\text{John})$$

$$\text{Greedy}(\text{John})$$

$$\text{Brother}(\text{Richard}, \text{John})$$

Instancjując zdania z kwantyfikatorami uniwersalnymi na *wszystkie możliwe* sposoby otrzymujemy

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

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

$$\text{King}(\text{John})$$

$$\text{Greedy}(\text{John})$$

$$\text{Brother}(\text{Richard}, \text{John})$$

Nowa KB jest sprowadzona do *języka zdań*: symbole zdaniowe to

$$\text{King}(\text{John}), \text{Greedy}(\text{John}), \text{Evil}(\text{John}), \text{King}(\text{Richard}) \text{ itd.}$$

Redukcja do rachunku zdań

Fakt: zdanie ustalone jest logiczną konsekwencją nowej KB

wtw

jest logiczną konsekwencją oryginalnej KB

Fakt: każda baza wiedzy może być sprowadzona do języka zdań
tak, żeby zachować logiczne konsekwencje

Pomysł:

sprowadź KB i zapytanie do języka zdań

zastosuj rezolucję

zwróć wynik

Redukcja do rachunku zdań

Problem: symbole funkcyjne, jest nieskończenie wiele termów,
np. $Father(Father(Father(John)))$

Tw: Herbrand (1930). Jeśli zdanie α jest logiczną konsekwencją KB w logice I rzędu to jest logiczną konsekwencją *skończonego* podzbioru KB sprowadzonej do języka zdań

Rozw: Dla $n = 0$ do ∞
sprowadź KB do języka zdań z termami do głębokości n
sprawdź, czy α jest logiczną konsekwencją nowej KB

Problem: działa, jeśli α jest logiczną konsekwencją, pętli się, jeśli nie jest

Twierdzenie: Turing (1936), Church (1936)
logiczna konsekwencja I rzędu jest *półrozstrzygalna*

Redukcja do rachunku zdań: efektywnosc

Redukcja do rachunku zdań generuje wiele nieistotnych zdań.

Np.

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

$$\text{King}(\text{John})$$

$$\forall y \text{ Greedy}(y)$$

$$\text{Brother}(\text{Richard}, \text{John})$$

fakt $\text{Evil}(\text{John})$ wydaje się oczywisty, ale redukcja produkuje wiele faktów takich jak $\text{Greedy}(\text{Richard})$ które są nieistotne

Dla p k -arnych predykatów i n stałych, będzie $p \cdot n^k$ instancjacji!

Unifikacja

Wniosek można otrzymać bezpośrednio, jeśli znajdziemy podstawienie θ takie, że $King(x)$ i $Greedy(x)$ pasują do $King(John)$ i $Greedy(y)$

$\theta = \{x/John, y/John\}$ spełnia te wymagania

$UNIFY(\alpha, \beta) = \theta$ jeśli $\alpha\theta = \beta\theta$

p	q	θ
$Knows(John, x)$	$Knows(John, Jane)$	
$Knows(John, x)$	$Knows(y, OJ)$	
$Knows(John, x)$	$Knows(y, Mother(y))$	
$Knows(John, x)$	$Knows(x, OJ)$	

Unifikacja

Wniosek można otrzymać bezpośrednio, jeśli znajdziemy podstawienie θ takie, że $King(x)$ i $Greedy(x)$ pasują do $King(John)$ i $Greedy(y)$

$\theta = \{x/John, y/John\}$ spełnia te wymagania

$UNIFY(\alpha, \beta) = \theta$ jeśli $\alpha\theta = \beta\theta$

p	q	θ
$Knows(John, x)$	$Knows(John, Jane)$	$\{x/Jane\}$
$Knows(John, x)$	$Knows(y, OJ)$	
$Knows(John, x)$	$Knows(y, Mother(y))$	
$Knows(John, x)$	$Knows(x, OJ)$	

Unifikacja

Wniosek można otrzymać bezpośrednio, jeśli znajdziemy podstawienie θ takie, że $King(x)$ i $Greedy(x)$ pasują do $King(John)$ i $Greedy(y)$

$\theta = \{x/John, y/John\}$ spełnia te wymagania

$UNIFY(\alpha, \beta) = \theta$ jeśli $\alpha\theta = \beta\theta$

p	q	θ
$Knows(John, x)$	$Knows(John, Jane)$	$\{x/Jane\}$
$Knows(John, x)$	$Knows(y, OJ)$	$\{x/OJ, y/John\}$
$Knows(John, x)$	$Knows(y, Mother(y))$	
$Knows(John, x)$	$Knows(x, OJ)$	

Unifikacja

Wniosek można otrzymać bezpośrednio, jeśli znajdziemy podstawienie θ takie, że $King(x)$ i $Greedy(x)$ pasują do $King(John)$ i $Greedy(y)$

$\theta = \{x/John, y/John\}$ spełnia te wymagania

$UNIFY(\alpha, \beta) = \theta$ jeśli $\alpha\theta = \beta\theta$

p	q	θ
$Knows(John, x)$	$Knows(John, Jane)$	$\{x/Jane\}$
$Knows(John, x)$	$Knows(y, OJ)$	$\{x/OJ, y/John\}$
$Knows(John, x)$	$Knows(y, Mother(y))$	$\{y/John, x/Mother(John)\}$
$Knows(John, x)$	$Knows(x, OJ)$	

Unifikacja

Wniosek można otrzymać bezpośrednio, jeśli znajdziemy podstawienie θ takie, że $King(x)$ i $Greedy(x)$ pasują do $King(John)$ i $Greedy(y)$

$\theta = \{x/John, y/John\}$ spełnia te wymagania

$UNIFY(\alpha, \beta) = \theta$ jeśli $\alpha\theta = \beta\theta$

p	q	θ
$Knows(John, x)$	$Knows(John, Jane)$	$\{x/Jane\}$
$Knows(John, x)$	$Knows(y, OJ)$	$\{x/OJ, y/John\}$
$Knows(John, x)$	$Knows(y, Mother(y))$	$\{y/John, x/Mother(John)\}$
$Knows(John, x)$	$Knows(x, OJ)$	<i>fail</i>

Uogólnione Modus Ponens

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

gdzie $p_i'\theta = p_i\theta$ dla wszyst. i

p_1' jest *King(John)* p_1 jest *King(x)*
 p_2' jest *Greedy(y)* p_2 jest *Greedy(x)*
 θ jest $\{x/\text{John}, y/\text{John}\}$ q jest *Evil(x)*
 $q\theta$ jest *Evil(John)*

Uogólnione Modus Ponens używa baz wiedzy klauzul definiujących (*dokładnie* jeden literał pozytywny)

Zakłada się, że wszystkie zmienne są kwantyfikowane uniwersalnie

Uogolnione Modus Ponens: poprawnosc

Trzeba pokazać, że

$$p_1', \dots, p_n', (p_1 \wedge \dots \wedge p_n \Rightarrow q) \models q\theta$$

przy założeniu, że $p_i'\theta = p_i\theta$ dla wszystkich i

Lemat: Dla dowolnej klauzuli definiującej p , mamy, że $p \models p\theta$ przez uniwersalną instancjację

1. $(p_1 \wedge \dots \wedge p_n \Rightarrow q) \models (p_1 \wedge \dots \wedge p_n \Rightarrow q)\theta = (p_1\theta \wedge \dots \wedge p_n\theta \Rightarrow q\theta)$
2. $p_1', \dots, p_n' \models p_1' \wedge \dots \wedge p_n' \models p_1'\theta \wedge \dots \wedge p_n'\theta$
3. Z 1 i 2 dostajemy, że $q\theta$ wynika na podstawie reguły Modus Ponens w rachunku zdań

Baza wiedzy: przykład

Prawo amerykańskie określa, że sprzedaż broni obcemu narodowi przez Amerykanina jest przestępstwem. Państwo Nono, które jest wrogiem, ma pewne pociski. Wszystkie pociski zostały mu sprzedane przez pułkownika Westa, który jest Amerykaninem.

Pokazać, że pułkownik West jest przestępcą.

Baza wiedzy: przykład

... sprzedaż broni obcemu narodowi przez Amerykanina jest przestępstwem:

Baza wiedzy: przykład

... sprzedaż broni obcemu narodowi przez Amerykanina jest przestępstwem:

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

Nono ... ma pewne pociski

Baza wiedzy: przykład

... sprzedaż broni obcemu narodowi przez Amerykanina jest przestępstwem:

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

Nono ... ma pewne pociski, tzn. $\exists x \textit{Owns}(\textit{Nono}, x) \wedge \textit{Missile}(x)$:

$$\textit{Owns}(\textit{Nono}, M_1) \textit{ i } \textit{Missile}(M_1)$$

... wszystkie pociski zostały mu sprzedane przez pułkownika Westa

Baza wiedzy: przykład

... sprzedaż broni obcemu narodowi przez Amerykanina jest przestępstwem:

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

Nono ... ma pewne pociski, tzn. $\exists x \textit{Owns}(\textit{Nono}, x) \wedge \textit{Missile}(x)$:

$$\textit{Owns}(\textit{Nono}, M_1) \textit{ i } \textit{Missile}(M_1)$$

... wszystkie pociski zostały mu sprzedane przez pułkownika Westa

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

Pociski są bronią:

Baza wiedzy: przykład

... sprzedaż broni obcemu narodowi przez Amerykanina jest przestępstwem:

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

Nono ... ma pewne pociski, tzn. $\exists x \textit{Owns}(\textit{Nono}, x) \wedge \textit{Missile}(x)$:

$$\textit{Owns}(\textit{Nono}, M_1) \textit{ i } \textit{Missile}(M_1)$$

... wszystkie pociski zostały mu sprzedane przez pułkownika Westa

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

Pociski są bronią:

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

Nieprzyjaciel Ameryki uznawany jest za "wrogi":

Baza wiedzy: przykład

... sprzedaż broni obcemu narodowi przez Amerykanina jest przestępstwem:

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

Nono ... ma pewne pociski, tzn. $\exists x \textit{Owns}(\textit{Nono}, x) \wedge \textit{Missile}(x)$:

$$\textit{Owns}(\textit{Nono}, M_1) \textit{ i } \textit{Missile}(M_1)$$

... wszystkie pociski zostały mu sprzedane przez pułkownika Westa

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

Pociski są bronią:

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

Nieprzyjaciel Ameryki uznawany jest za "wrogi":

$$\textit{Enemy}(x, \textit{America}) \Rightarrow \textit{Hostile}(x)$$

West, który jest Amerykaninem ...

$$\textit{American}(\textit{West})$$

Państwo Nono, nieprzyjaciel Ameryki ...

$$\textit{Enemy}(\textit{Nono}, \textit{America})$$

Forward chaining: algorytm

```
function FOL-FC-ASK( $KB, \alpha$ ) returns a substitution or false  
repeat until new is empty  
   $new \leftarrow \{ \}$   
  for each sentence  $r$  in  $KB$  do  
     $(p_1 \wedge \dots \wedge p_n \Rightarrow q) \leftarrow \text{STANDARDIZE-APART}(r)$   
    for each  $\theta$  such that  $(p_1 \wedge \dots \wedge p_n)\theta = (p'_1 \wedge \dots \wedge p'_n)\theta$   
      for some  $p'_1, \dots, p'_n$  in  $KB$   
         $q' \leftarrow \text{SUBST}(\theta, q)$   
        if  $q'$  is not a renaming of a sentence already in  $KB$  or new then do  
          add  $q'$  to new  
           $\phi \leftarrow \text{UNIFY}(q', \alpha)$   
          if  $\phi$  is not fail then return  $\phi$   
  add new to  $KB$   
return false
```

Forward chaining: przykład

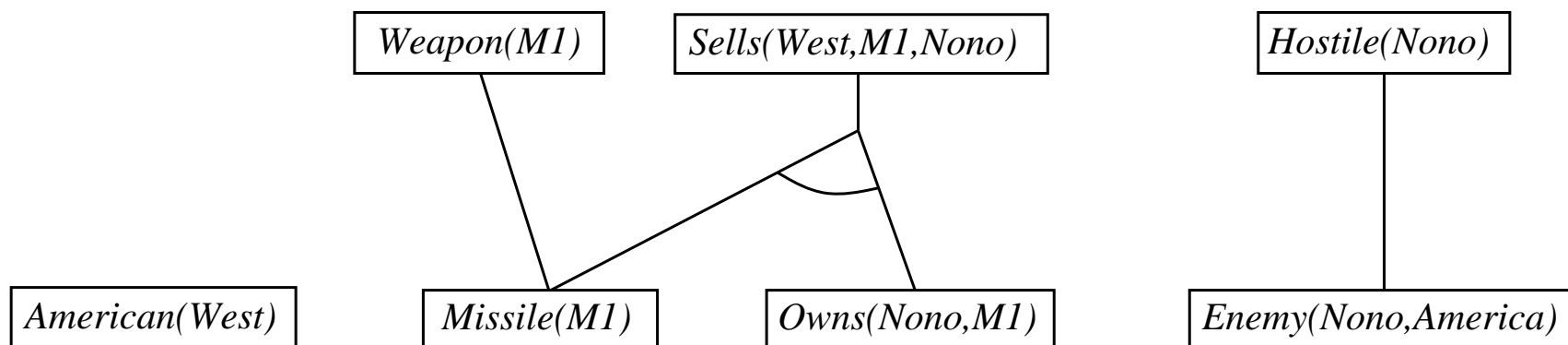
American(West)

Missile(M1)

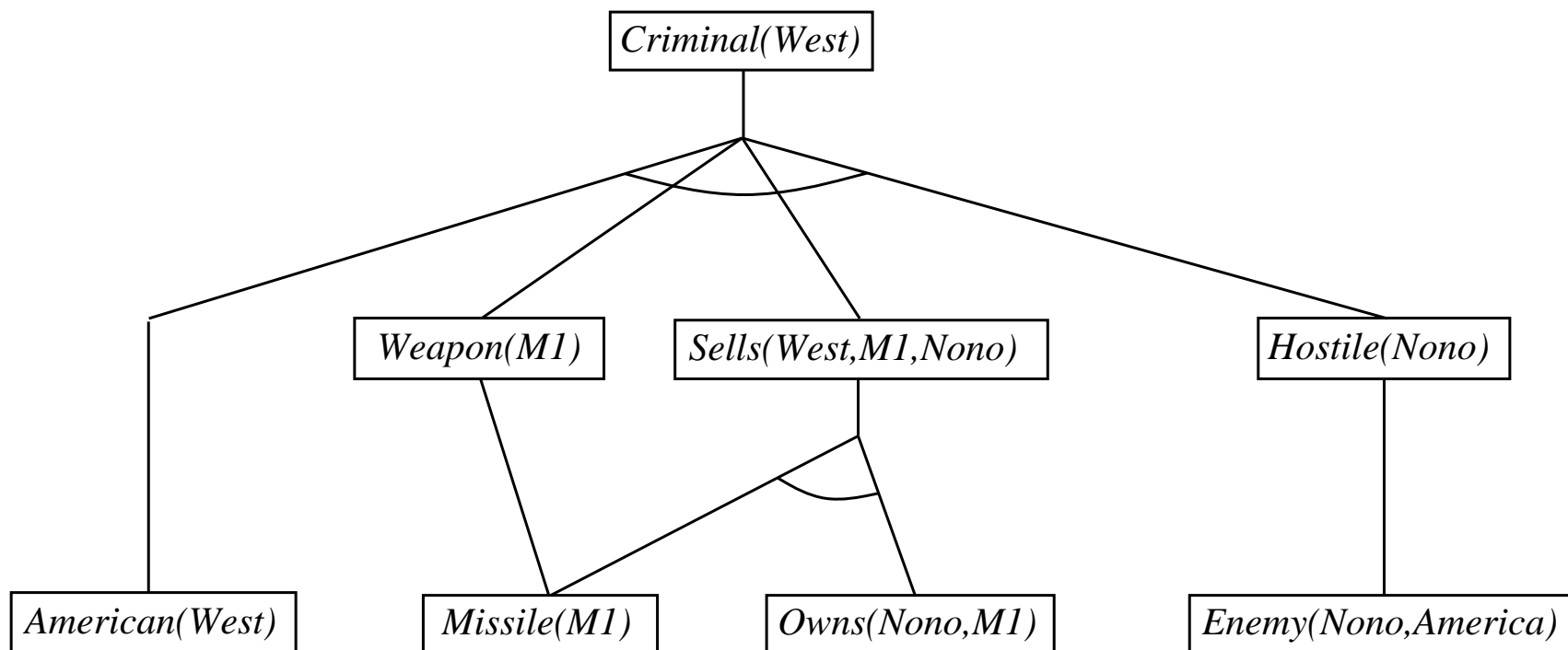
Owns(Nono,M1)

Enemy(Nono,America)

Forward chaining: przykład



Forward chaining: przykład



Forward chaining: własności

Poprawny i pełny dla baz wiedzy w postaci klauzul definiujących I rzędu
(dowód analogiczny do dowodu dla rachunku zdań)

Datalog = klauzule definiujące I rzędu + *brak funkcji*

Forward chaining dla Datalogu kończy się zawsze po wielomianowej liczbie iteracji: co najwyżej $p \cdot n^k$ literałów

W ogólności może się zapętlić jeśli α nie jest logiczną konsekwencją

To jest nieuniknione: logiczna konsekwencja z klauzulami definiującymi jest półrozstrzygalna

Forward chaining: efektywnosc

Prosta obserwacja: nie trzeba dopasowywać reguły w k -tej iteracji jeśli przesłanka reguły nie została dodana w $k - 1$ -szej iteracji

⇒ dopasowuj każdą regułę, której przesłanki zawierają nowo dodany literał

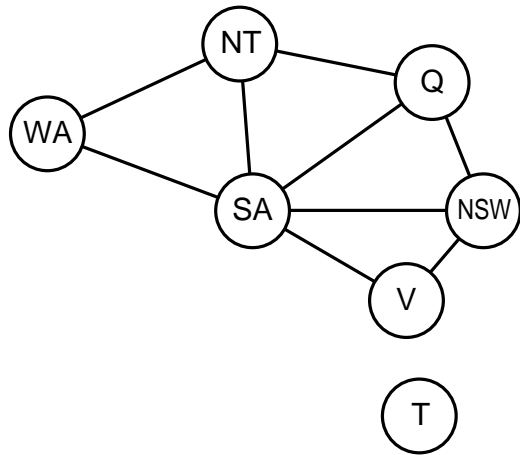
Dopasowywanie może być kosztowne

Indeksowanie faktów umożliwia sprawdzenie znanych faktów w czasie $O(1)$, ale nie rozwiązuje problemu dla częściowo ustalonych zapytań

W ogólności dopasowywanie koniunkcji przesłanek do znanych faktów jest NP-trudne

Forward chaining jest powszechnie stosowany w dedukcyjnych bazach danych

Forward chaining: NP-trudnosc dopasowywania



$$\begin{aligned}
 & Diff(wa, nt) \wedge Diff(wa, sa) \wedge \\
 & Diff(nt, q) Diff(nt, sa) \wedge \\
 & Diff(q, nsw) \wedge Diff(q, sa) \wedge \\
 & Diff(nsw, v) \wedge Diff(nsw, sa) \wedge \\
 & Diff(v, sa) \Rightarrow Colorable()
 \end{aligned}$$

$$\begin{aligned}
 & Diff(Red, Blue) \quad Diff(Red, Green) \\
 & Diff(Green, Red) \quad Diff(Green, Blue) \\
 & Diff(Blue, Red) \quad Diff(Blue, Green)
 \end{aligned}$$

$Colorable()$ wynika wtw CSP ma rozwiązanie

Problemy CSP zawierają 3SAT jako szczególny przypadek, stąd dopasowywanie jest NP-trudne

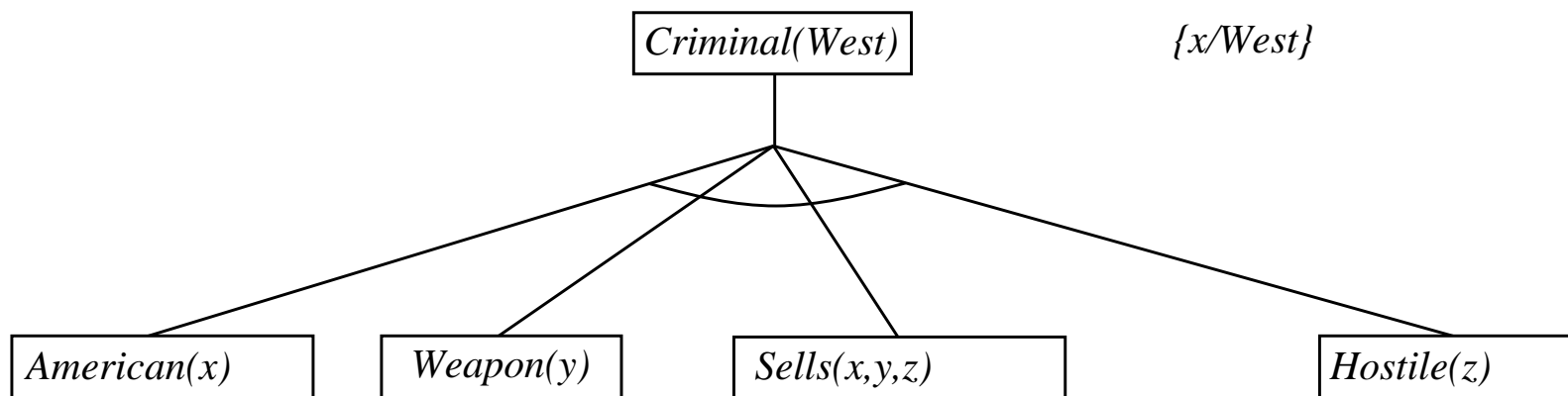
Backward chaining: algorytm

```
function FOL-BC-ASK(KB, goals,  $\theta$ ) returns a set of substitutions
  inputs: KB, a knowledge base
            goals, a list of conjuncts forming a query
             $\theta$ , the current substitution, initially the empty substitution { }
  local variables: ans, a set of substitutions, initially empty
  if goals is empty then return { $\theta$ }
   $q' \leftarrow \text{SUBST}(\theta, \text{FIRST}(\textit{goals}))$ 
  for each r in KB where  $\text{STANDARDIZE-APART}(r) = (p_1 \wedge \dots \wedge p_n \Rightarrow q)$ 
    and  $\theta' \leftarrow \text{UNIFY}(q, q')$  succeeds
     $\textit{ans} \leftarrow \text{FOL-BC-ASK}(\textit{KB}, [p_1, \dots, p_n | \text{REST}(\textit{goals})], \text{COMPOSE}(\theta', \theta)) \cup \textit{ans}$ 
  return ans
```

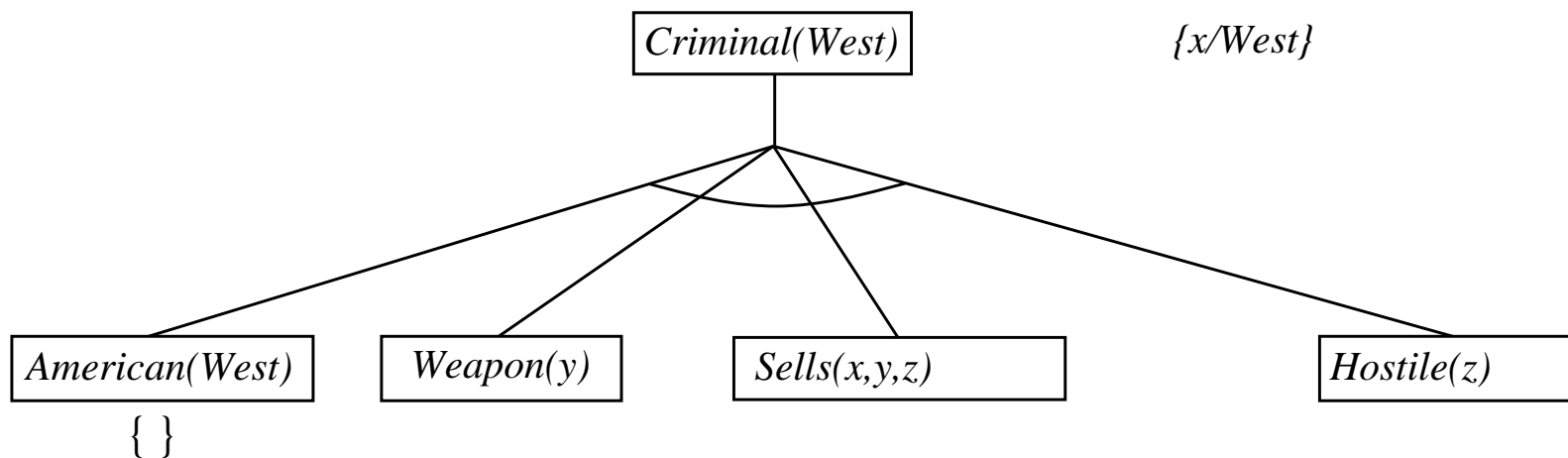
Backward chaining: przykład

Criminal(West)

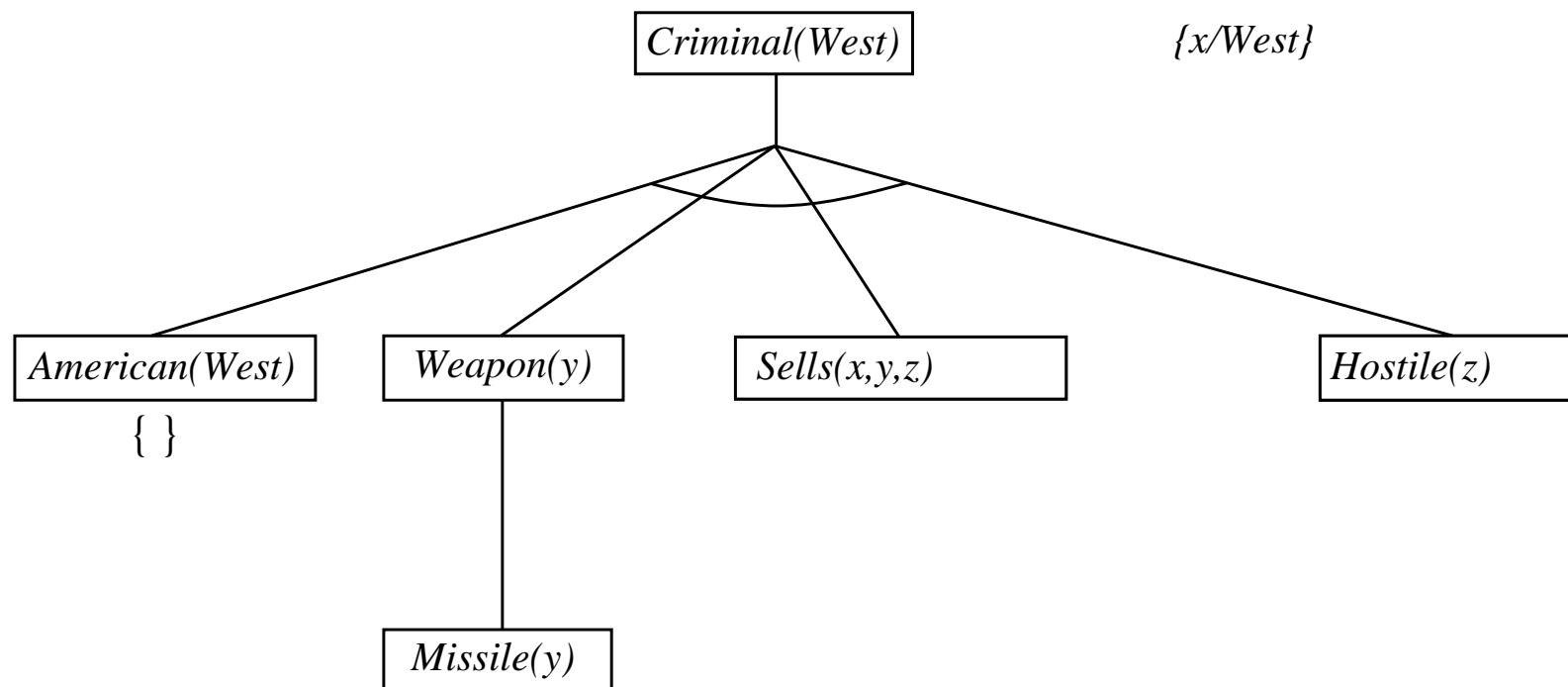
Backward chaining: przykład



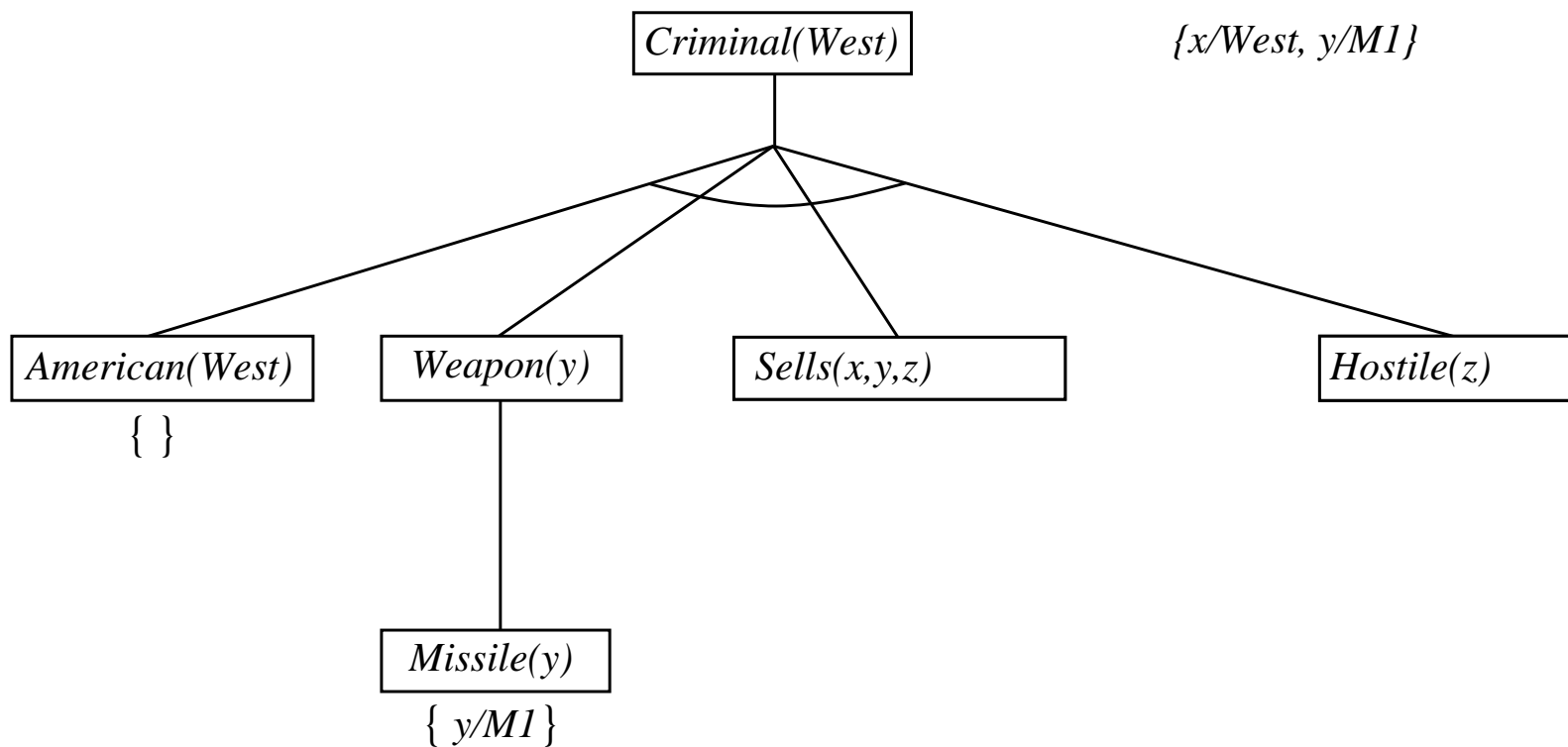
Backward chaining: przykład



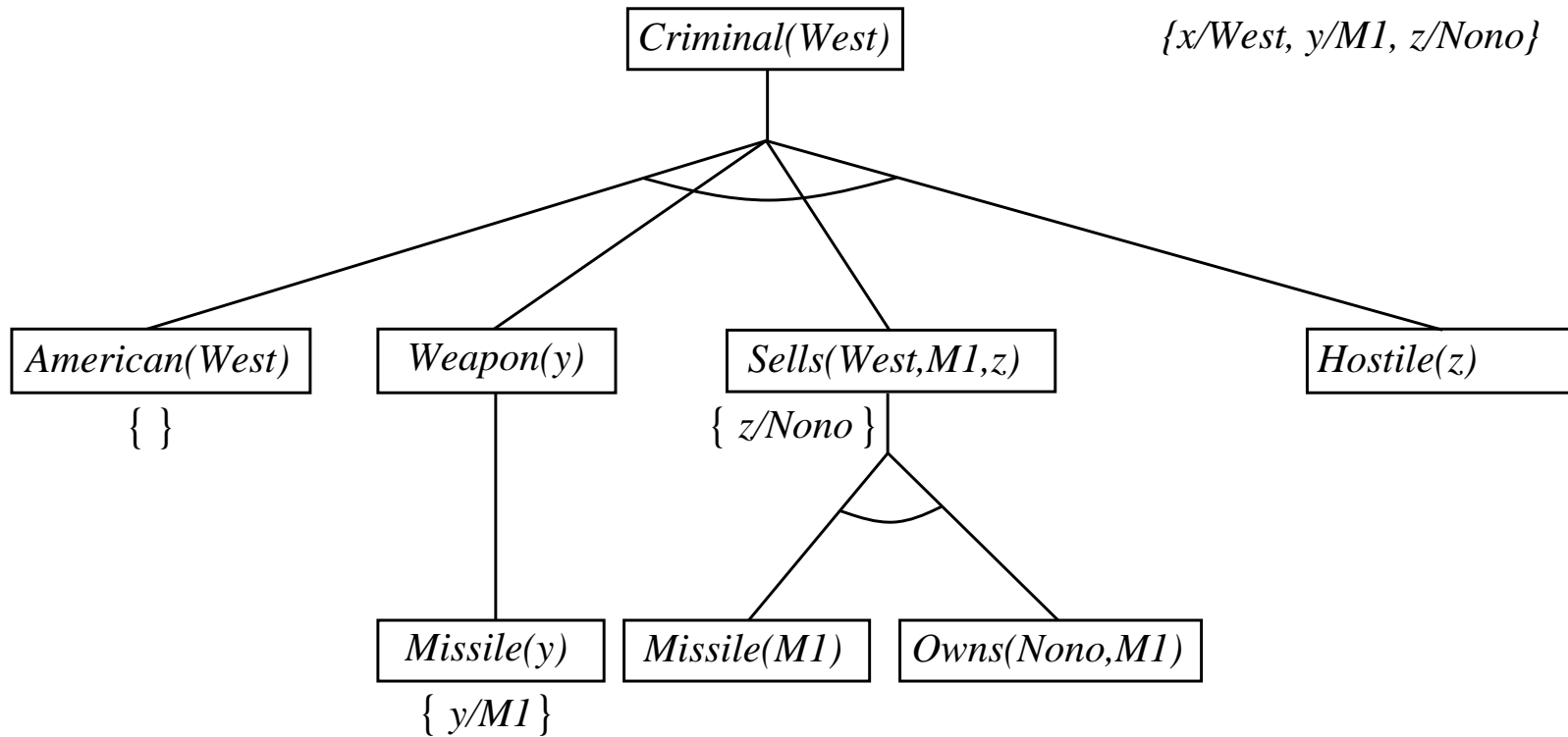
Backward chaining: przykład



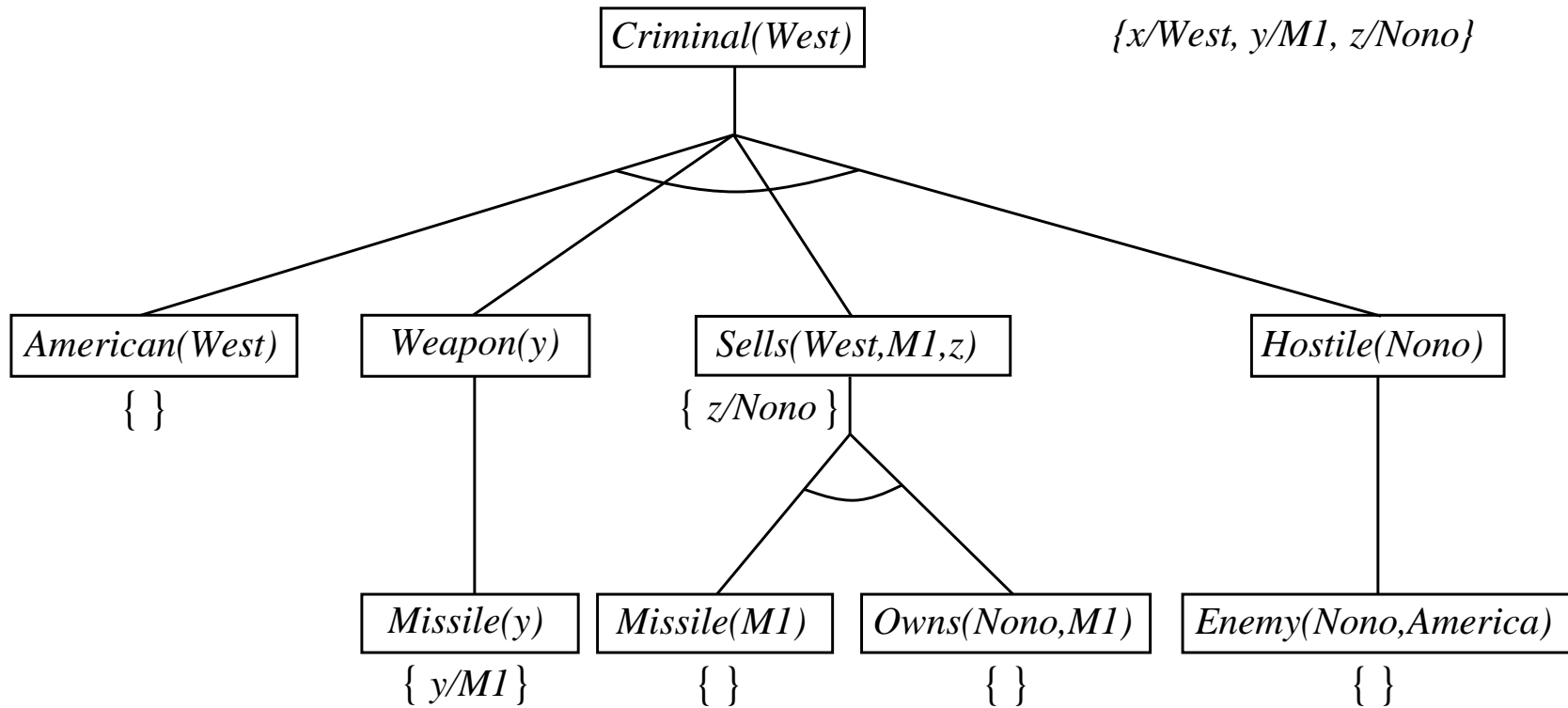
Backward chaining: przykład



Backward chaining: przykład



Backward chaining: przykład



Backward chaining: własności

Przeszukiwanie rekurencyjne w głąb: pamięć jest liniowa od rozmiaru dowodu

Niepełny z powodu zapętleń

⇒ można poprawić sprawdzając każdy nowo wygenerowany cel, czy nie wystąpił już wcześniej

Nieefektywny z powodu powtarzających się celów (udowodnionych i porażek)

⇒ można poprawić zapamiętując wcześniejsze wyniki (wymaga dodatkowej pamięci!)

Powszechnie stosowany w [programowaniu w logice](#) (Prolog)

Rezolucja

Reguła rezolucji dla logiki I rzędu:

$$\frac{l_1 \vee \dots \vee l_k, \quad m_1 \vee \dots \vee m_n}{(l_1 \vee \dots \vee l_{i-1} \vee l_{i+1} \vee \dots \vee l_k \vee m_1 \vee \dots \vee m_{j-1} \vee m_{j+1} \vee \dots \vee m_n)\theta}$$

gdzie $\text{UNIFY}(l_i, \neg m_j) = \theta$.

Na przykład,

$$\frac{\neg Rich(x) \vee Unhappy(x) \\ Rich(Ken)}{Unhappy(Ken)}$$

z unifikacją $\theta = \{x/Ken\}$

Algorytm stosuje rezolucję do $CNF(KB \wedge \neg\alpha)$

Problem:

Rezolucja binarna jest poprawnym, ale nie jest pełnym systemem wnioskowania dla logiki I rzędu

Faktoryzacja

Reguła faktoryzacji dla logiki I rzędu (usuwanie powtarzających się literałów):

$$\frac{l_1 \vee \dots \vee l_i \vee \dots \vee l_j \vee \dots \vee l_k}{(l_1 \vee \dots \vee l_i \vee \dots \vee l_{j-1} \vee l_{j+1} \vee \dots \vee l_k)\theta}$$

gdzie $\text{UNIFY}(l_i, l_j) = \theta$.

Na przykład,

$$\frac{\text{Likes}(x, \text{Chris}) \vee \text{Likes}(\text{Ken}, y)}{\text{Likes}(\text{Ken}, \text{Chris})}$$

z unifikacją $\theta = \{x/\text{Ken}, y/\text{Chris}\}$

Rezolucja + faktoryzacja jest poprawnym i pełnym systemem wnioskowania dla logiki I rzędu (bez ograniczeń)

Rezolucja: przekształcanie do CNF

Każdy, kto kocha wszystkie zwierzęta, jest kochany przez kogoś:

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

1. Eliminacja równoważności i implikacji

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

2. Przemieszczenie \neg do środka: $\neg \forall x, p \equiv \exists x \neg p$, $\neg \exists x, p \equiv \forall x \neg p$:

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

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

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

Rezolucja: przekształcanie do CNF

3. Przemianowanie zmiennych tak, żeby każdy kwantyfikator miał inną

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

4. Skolemizacja: bardziej ogólna postać egzystencjalnej instancjacji.

Każda zmienna kwantyfikowana egzystencjalnie jest zastępowana przez funkcję Skolema otaczających ją uniwersalnie kwantyf. zmiennych:

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

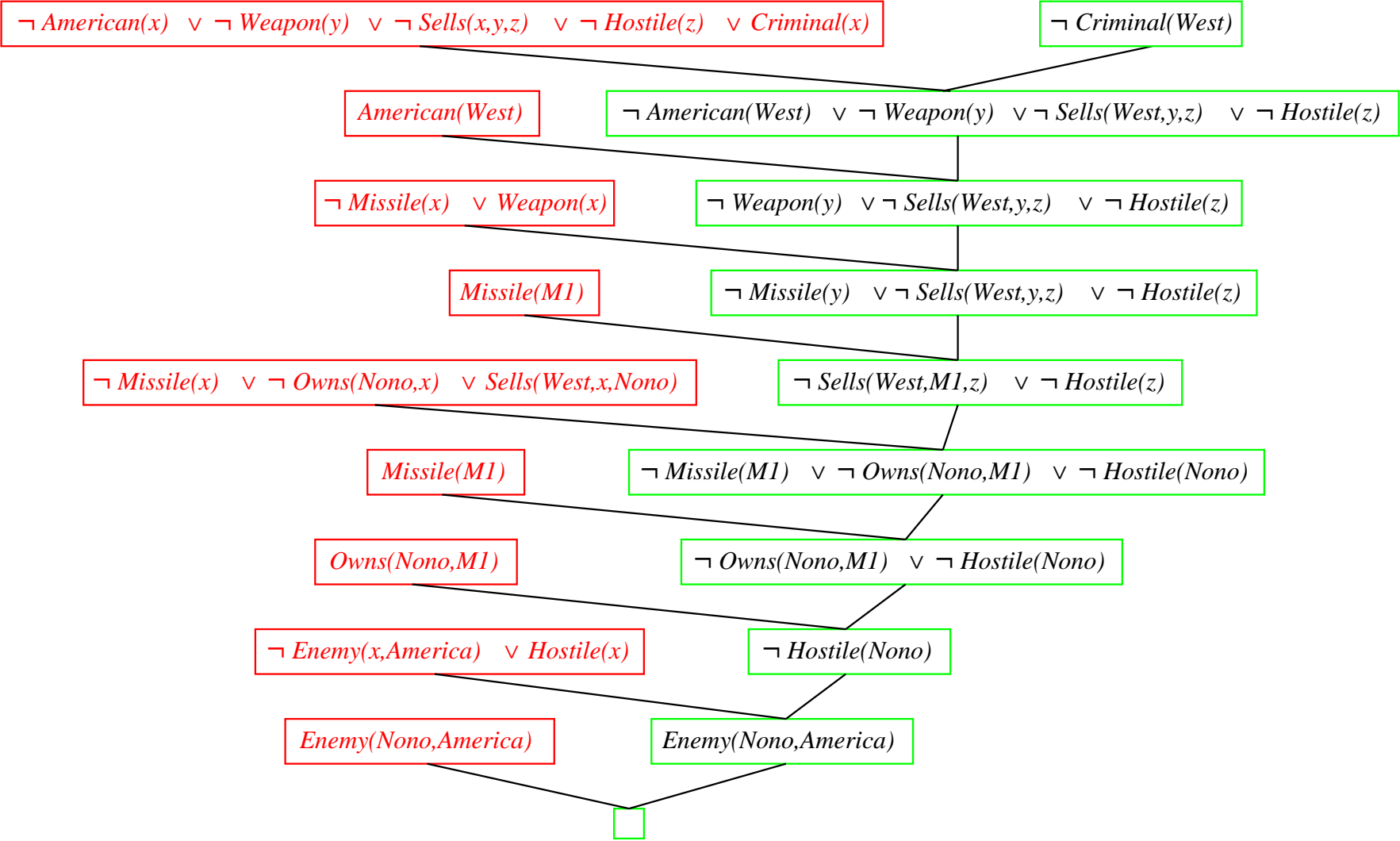
5. Usunięcie kwantyfikatorów uniwersalnych:

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

6. Rozdzielenie \wedge względem \vee :

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

Rezolucja: przykład



Systemy wnioskowania: podsumowanie

Język	Ontologia	Epistemologia
Rachunek zdań	fakty	prawda/fałsz/nieznane
Logika I rzędu	fakty, obiekty, relacje	prawda/fałsz/nieznane
Logika temporalna	fakty, obiekty, relacje, czas	prawda/fałsz/nieznane
Prawdopodobieństwo	fakty	prawdopodobieństwo $\in [0, 1]$
Logika rozmyta	stopień prawdziwości $\in [0, 1]$	przedziały $\subset [0, 1]$