Sztuczna Inteligencja i Systemy Doradcze

Wnioskowanie w logice I rzędu

Rachunek zdan: 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 rzedu

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

Syntaktyka: zdania atomowe

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

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

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

Syntaktyka: zdania zlozone

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

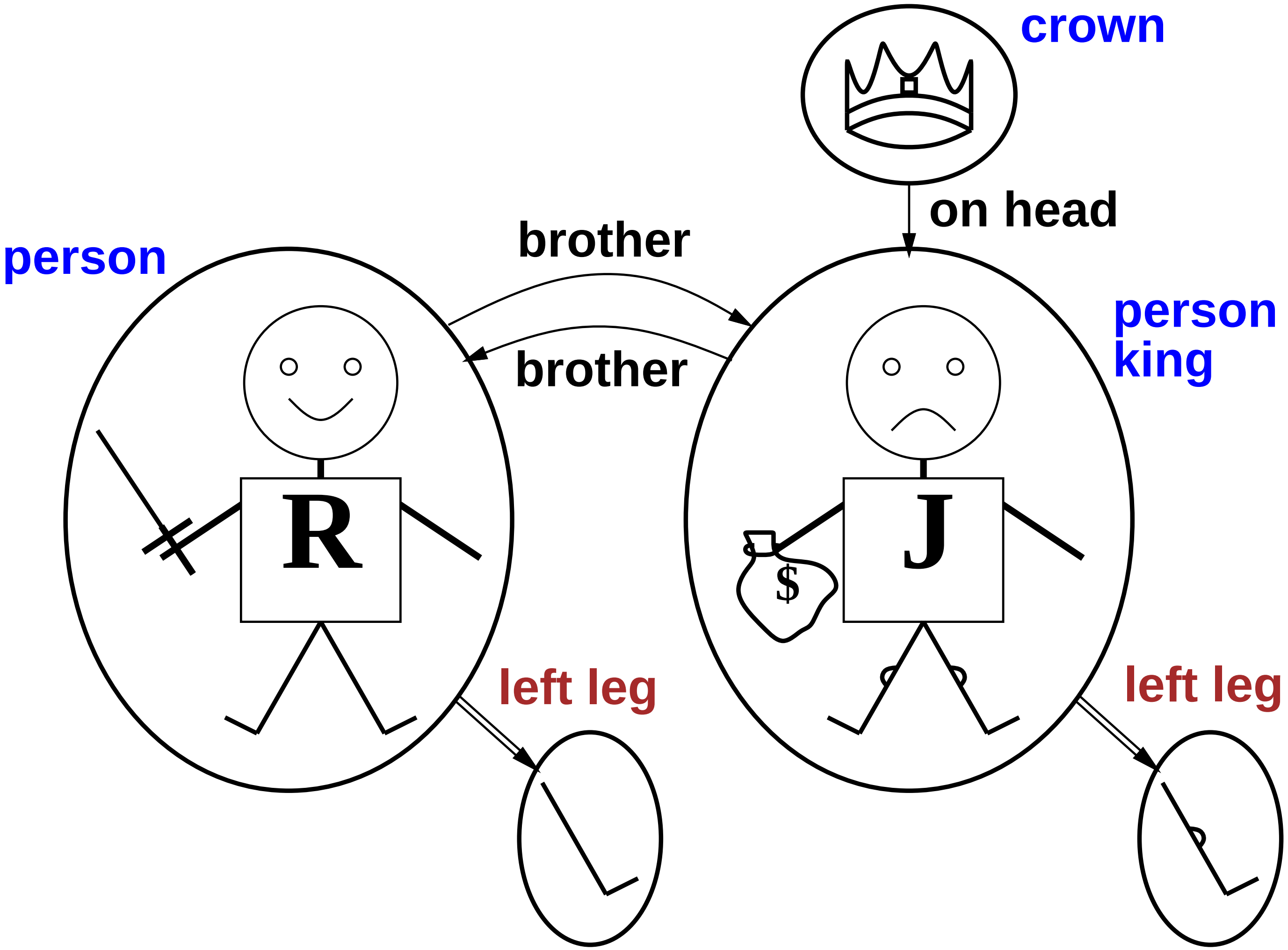
Model zawiera ≥ 1 obiektów (elementów dziedziny) i relacje między nimi

Interpretacja specyfikuje przyporządkowania:

symbole stałe $\longrightarrow$ obiekty
predykaty $\longrightarrow$ relacje
symbole funkcyjne $\longrightarrow$ relacje funkcyjne

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

Model: przyklad



Modele logiki I rzedu: bardzo duzo!

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:
$\forall x \; At(x, Berkeley) \Rightarrow Smart(x)$

$\forall x \; P$ 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

$$\begin{aligned} & At(KingJohn, Berkeley) \Rightarrow Smart(KingJohn) \\ \wedge \; & At(Richard, Berkeley) \Rightarrow Smart(Richard) \\ \wedge \; & At(Berkeley, Berkeley) \Rightarrow Smart(Berkeley) \\ \wedge \; & \ldots \end{aligned}$$

Kwantyfikatory egzystencjalne

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

Ktoś w Stanford jest sprytny:
$\exists x \;\; At(x, Stanford) \land Smart(x)$

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

Nieformalnie, równoważne alternatywie podstawień w P

$$\begin{array}{ll} & At(KingJohn, Stanford) \land Smart(KingJohn) \\ \lor & At(Richard, Stanford) \land Smart(Richard) \\ \lor & At(Stanford, Stanford) \land Smart(Stanford) \\ \lor & \ldots \end{array}$$

Kwantyfikatory: wlasnosci

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

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

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

$\exists x \ \ \forall y \ \ Loves(x, y)$
"Istnieje osoba, która kocha wszystkich"

$\forall y \ \ \exists x \ \ Loves(x, y)$
"Każdy jest kochany przez co najmniej jedną osobę"

Dualność kwantyfikatorów: jeden można opisać przy pomocy drugiego

$\forall x \ \ Likes(x, IceCream) \qquad \neg\exists x \ \ \neg Likes(x, IceCream)$

$\exists x \ \ Likes(x, Broccoli) \qquad \neg\forall x \ \ \neg Likes(x, Broccoli)$

Tlumaczenie jezyka naturalnego do I rzedu

Bracia są rodzeństwem

Tlumaczenie jezyka naturalnego do I rzedu

Bracia są rodzeństwem

$\forall\, x, y \;\; Brother(x, y) \;\Rightarrow\; Sibling(x, y)$.

"Rodzeństwo" jest symetryczne

Tlumaczenie jezyka naturalnego do I rzedu

Bracia są rodzeństwem

$\forall\, x, y \;\; Brother(x, y) \;\Rightarrow\; Sibling(x, y)$.

"Rodzeństwo" jest symetryczne

$\forall\, x, y \;\; Sibling(x, y) \;\Leftrightarrow\; Sibling(y, x)$.

Czyjaś matka jest kobietą i jego rodzicem

Tlumaczenie jezyka naturalnego do I rzedu

Bracia są rodzeństwem

$\forall\, x, y \;\; Brother(x, y) \;\Rightarrow\; Sibling(x, y)$.

"Rodzeństwo" jest symetryczne

$\forall\, x, y \;\; Sibling(x, y) \;\Leftrightarrow\; Sibling(y, x)$.

Czyjaś matka jest kobietą i jego rodzicem

$\forall\, x, y \;\; Mother(x, y) \;\Leftrightarrow\; (Female(x) \wedge Parent(x, y))$.

Kuzyn jest dzieckiem rodzeństwa rodzica

Tlumaczenie jezyka naturalnego do I rzedu

Bracia są rodzeństwem

$\forall\, x, y \;\; Brother(x, y) \;\Rightarrow\; Sibling(x, y).$

"Rodzeństwo" jest symetryczne

$\forall\, x, y \;\; Sibling(x, y) \;\Leftrightarrow\; Sibling(y, x).$

Czyjaś matka jest kobietą i jego rodzicem

$\forall\, x, y \;\; Mother(x, y) \;\Leftrightarrow\; (Female(x) \wedge Parent(x, y)).$

Kuzyn jest dzieckiem rodzeństwa rodzica

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

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 $Parent$:

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

Baza wiedzy w jezyku I rzedu

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, $\{a/Shoot\}$ $\leftarrow$ 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: przyklad

"Obserwacja"

$\forall\, b, g, t\;\; Percept([Smell, b, g], t) \;\Rightarrow\; Smelt(t)$

$\forall\, s, b, t\;\; Percept([s, b, Glitter], t) \;\Rightarrow\; AtGold(t)$

Reakcja: $\forall\, t\;\; AtGold(t) \;\Rightarrow\; Action(Grab, t)$

Reakcja z wewnętrznym stanem: czy nie mamy już złota?

$\forall\, t\;\; AtGold(t) \wedge \neg Holding(Gold, t) \;\Rightarrow\; Action(Grab, t)$

Baza wiedzy: przyklad

Własności miejsca:

$\forall\, x, t \;\; At(Agent, x, t) \land Smelt(t) \;\Rightarrow\; Smelly(x)$

$\forall\, x, t \;\; At(Agent, x, t) \land Breeze(t) \;\Rightarrow\; Breezy(x)$

Blisko pułapek jest wiatr:

Reguła diagnostyczna — wnioskuje przyczynę z efektu

$$\forall\, y \;\; Breezy(y) \;\Rightarrow\; \exists\, x \;\; Pit(x) \land Adjacent(x, y)$$

Reguła przyczynowa — wnioskuje efekt z przyczyny

$$\forall\, x, y \;\; Pit(x) \land Adjacent(x, y) \;\Rightarrow\; Breezy(y)$$

Definicja: $\forall\, y \;\; Breezy(y) \;\Leftrightarrow\; [\exists\, x \;\; Pit(x) \land Adjacent(x, y)]$

Baza wiedzy: przyklad

Własności miejsca:
$\forall\, x, t \;\; At(Agent, x, t) \land Smelt(t) \;\Rightarrow\; Smelly(x)$
$\forall\, x, t \;\; At(Agent, x, t) \land Breeze(t) \;\Rightarrow\; Breezy(x)$

Blisko pułapek jest wiatr:

Reguła diagnostyczna — wnioskuje przyczynę z efektu

$$\forall\, y \;\; Breezy(y) \;\Rightarrow\; \exists\, x \;\; Pit(x) \land Adjacent(x, y)$$

Reguła przyczynowa — wnioskuje efekt z przyczyny

$$\forall\, x, y \;\; Pit(x) \land Adjacent(x, y) \;\Rightarrow\; Breezy(y)$$

Definicja: $\forall\, y \;\; Breezy(y) \;\Leftrightarrow\; [\exists\, x \;\; Pit(x) \land Adjacent(x, y)]$

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

Wnioskowanie w logice I rzedu

♢ 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:

$$\frac{\forall\, v \;\; \alpha}{\textsc{Subst}(\{v/g\}, \alpha)}$$

dla dowolnej zmiennej v i termu ustalonego g

Np. $\forall\, x \;\; King(x) \land Greedy(x) \;\Rightarrow\; Evil(x)$ daje

$$\begin{array}{l}
King(John) \land Greedy(John) \;\Rightarrow\; Evil(John) \\
King(Richard) \land Greedy(Richard) \;\Rightarrow\; Evil(Richard) \\
King(Father(John)) \land Greedy(Father(John)) \;\Rightarrow\; Evil(Father(John)) \\
\quad \vdots
\end{array}$$

Instancjacja egzystencjalna (EI)

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

$$\frac{\exists\, v \;\; \alpha}{\textsc{Subst}(\{v/k\}, \alpha)}$$

Np. $\exists\, x \;\; Crown(x) \wedge OnHead(x, John)$ pociąga

$$Crown(C_1) \wedge OnHead(C_1, John)$$

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

Inny przykład: z $\exists\, x \;\; d(x^y)/dy = x^y$ otrzymujemy

$$d(e^y)/dy = e^y$$

przy założeniu, że e jest nowym symbolem stałym

Instancjacja: wlasnosci

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 zdan

Załóżmy, że mamy daną następującą bazę wiedzy KB:

$$\forall x \;\; King(x) \wedge Greedy(x) \;\Rightarrow\; Evil(x)$$
$$King(John)$$
$$Greedy(John)$$
$$Brother(Richard, John)$$

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

$$King(John) \wedge Greedy(John) \;\Rightarrow\; Evil(John)$$
$$King(Richard) \wedge Greedy(Richard) \;\Rightarrow\; Evil(Richard)$$
$$King(John)$$
$$Greedy(John)$$
$$Brother(Richard, John)$$

Nowa KB jest sprowadzona do języka zdań: symbole zdaniowe to

$King(John)$, $Greedy(John)$, $Evil(John)$, $King(Richard)$ itd.

Redukcja do rachunku zdan

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 zdan

Problem: symbole funkcyjne, jest nieskończnie 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 ∞
sprowadć 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 zdan: efektywnosc

Redukcja do rachunku zdań generuje wiele nieistotnych zdań.
Np.

$$\forall x \;\; King(x) \wedge Greedy(x) \;\Rightarrow\; Evil(x)$$
$$King(John)$$
$$\forall y \;\; Greedy(y)$$
$$Brother(Richard, John)$$

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

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

Unifikacja

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

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

$\text{UNIFY}(\alpha, \beta) = \theta$ 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 znajdziemiy podstawienie θ takie, że $King(x)$ i $Greedy(x)$ pasują do $King(John)$ i $Greedy(y)$

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

$\textsc{Unify}(\alpha, \beta) = \theta$ 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 znajdziemiy podstawienie θ takie, że $King(x)$ i $Greedy(x)$ pasują do $King(John)$ i $Greedy(y)$

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

$\textsc{Unify}(\alpha, \beta) = \theta$ 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 znajdziemiy podstawienie θ takie, że $King(x)$ i $Greedy(x)$ pasują do $King(John)$ i $Greedy(y)$

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

$\textsc{Unify}(\alpha, \beta) = \theta$ 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 znajdziemiy podstawienie θ takie, że $King(x)$ i $Greedy(x)$ pasują do $King(John)$ i $Greedy(y)$

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

$\textsc{Unify}(\alpha, \beta) = \theta$ jeśli $\alpha\theta = \beta\theta$

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

Uogolnione Modus Ponens

$$\frac{p_1{}',\;\; p_2{}',\; \ldots,\; p_n{}',\;\; (p_1 \land p_2 \land \ldots \land p_n \Rightarrow q)}{q\theta} \qquad \text{gdzie } p_i{}'\theta = p_i\theta \text{ dla wszyst. } i$$

$p_1{}'$ jest $King(John)$ $\qquad$ p_1 jest $King(x)$
$p_2{}'$ jest $Greedy(y)$ $\qquad$ p_2 jest $Greedy(x)$
θ jest $\{x/John, y/John\}$ $\qquad$ 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

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

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

Lemat: Dla dowolnej klauzuli deifniującej p, mamy, że $p \models p\theta$ przez uniwersalna instancjację

1. $(p_1 \wedge \ldots \wedge p_n \Rightarrow q) \models (p_1 \wedge \ldots \wedge p_n \Rightarrow q)\theta = (p_1\theta \wedge \ldots \wedge p_n\theta \Rightarrow q\theta)$

2. $p_1', \ \ldots, \ p_n' \models p_1' \wedge \ldots \wedge p_n' \models p_1'\theta \wedge \ldots \wedge p_n'\theta$

3. Z 1 i 2 dostajemy, że $q\theta$ wynika na podstawie reguły Modus Ponens w rachunku zdań

Baza wiedzy: przyklad

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

Pokazać, że pułkownik West jest przestępcą.

Baza wiedzy: przyklad

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

Baza wiedzy: przyklad

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

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

Nono ... ma pewne pociski

Baza wiedzy: przykład

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

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

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

$Owns(Nono, M_1)$ i $Missile(M_1)$

... wszystkie pociski zostały mu sprzedane przez pułkownika Westa

Baza wiedzy: przyklad

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

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

Nono ... ma pewne pociski, tzn. $\exists\, x\ Owns(Nono, x) \land Missile(x)$:

$Owns(Nono, M_1)$ i $Missile(M_1)$

... wszystkie pociski zostały mu sprzedane przez pułkownika Westa

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

Pociski są bronią:

Baza wiedzy: przyklad

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

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

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

$Owns(Nono, M_1)$ i $Missile(M_1)$

... wszystkie pociski zostały mu sprzedane przez pułkownika Westa

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

Pociski są bronią:

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

Nieprzyjaciel Ameryki uznawany jest za “wrogi”:

Baza wiedzy: przykład

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

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

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

$Owns(Nono,M_1)$ i $Missile(M_1)$

... wszystkie pociski zostały mu sprzedane przez pułkownika Westa

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

Pociski są bronią:

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

Nieprzyjaciel Ameryki uznawany jest za "wrogi":

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

West, który jest Amerykaninem ...

$American(West)$

Państwo Nono, nieprzyjaciel Ameryki ...

$Enemy(Nono,America)$

Forward chaining: algorytm

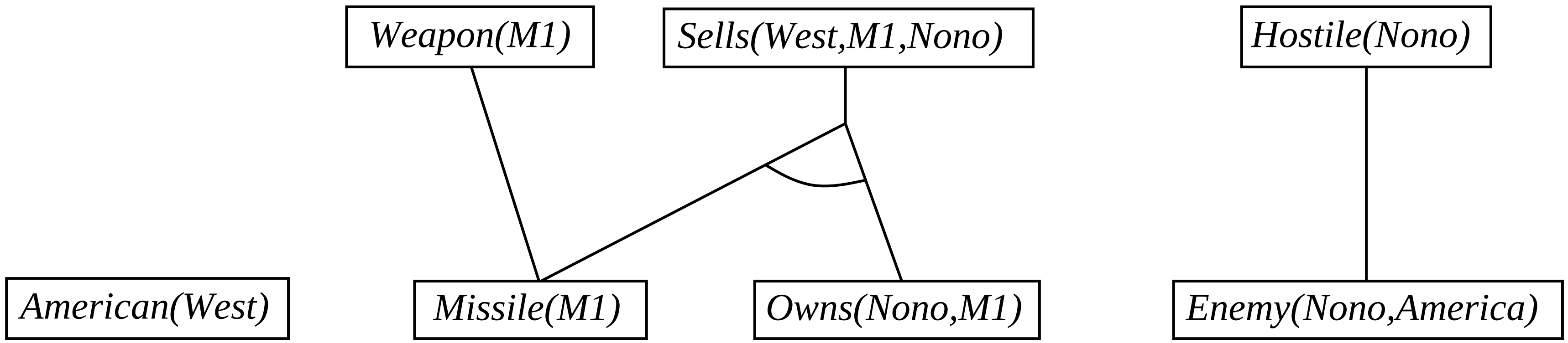
function FOL-FC-ASK(KB, α) **returns** a substitution or $false$

repeat until new is empty
    $new \leftarrow \{\,\}$
    for each sentence r **in** KB **do**
        $(\, p_1 \wedge \ldots \wedge\; p_n \;\Rightarrow\; q) \leftarrow$ STANDARDIZE-APART(r)
        for each θ such that $(p_1 \;\wedge\; \ldots \;\wedge\; p_n)\theta \;=\; (p'_1 \;\wedge\; \ldots \;\wedge\; p'_n)\theta$
                for some $p'_1, \ldots, p'_n$ in KB
            $q' \leftarrow$ SUBST(θ, q)
            if q' is not a renaming of a sentence already in KB or new **then do**
                add q' to new
                $\phi \leftarrow$ UNIFY(q', α)
                if ϕ is not $fail$ **then return** ϕ
    add new to KB
return $false$

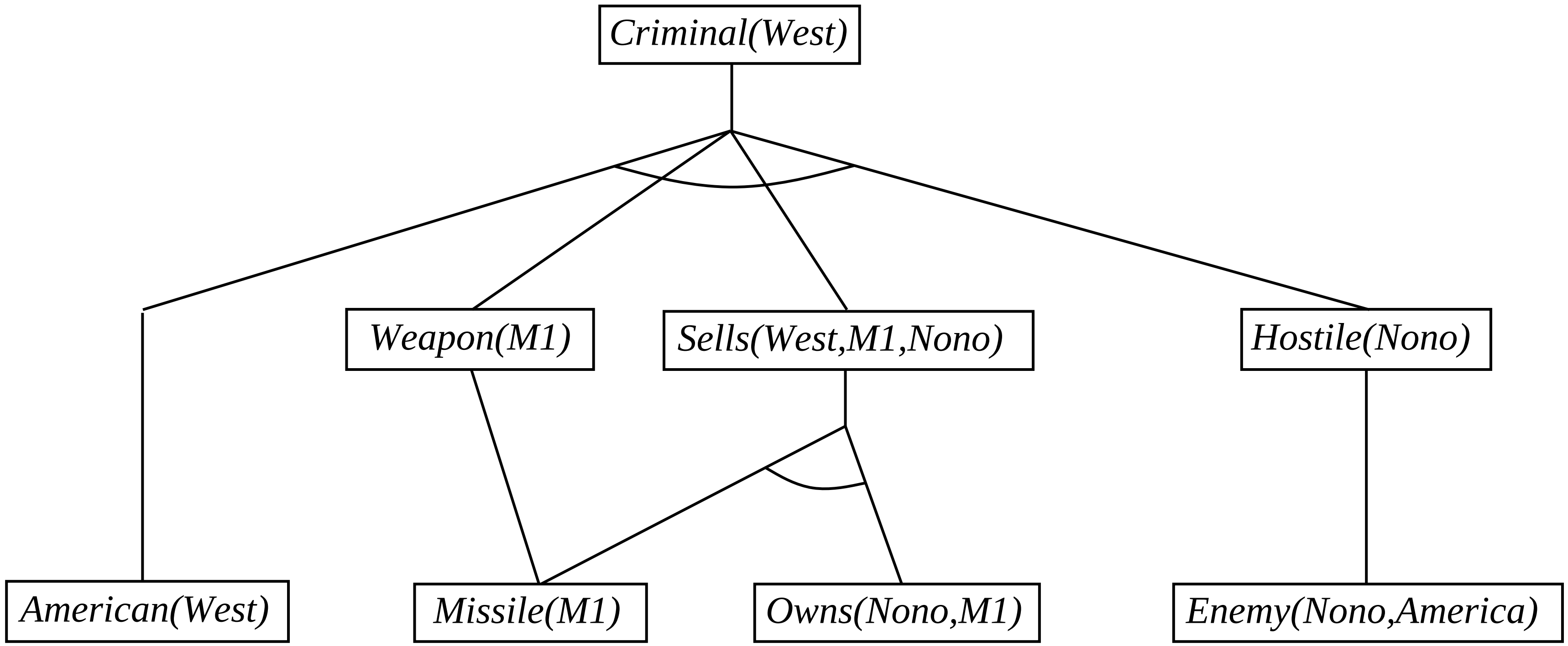
Forward chaining: przykład

American(West) *Missile(M1)* *Owns(Nono,M1)* *Enemy(Nono,America)*

Forward chaining: przyklad



Forward chaining: przyklad



Forward chaining: wlasnosci

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 logiczą 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

$\Rightarrow$ dopasowywuj każdą regułę, której przesłanki zawierają nowo dodany literał

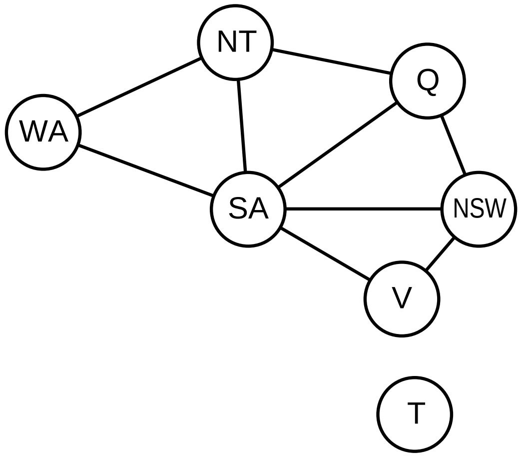
Dopasowywanie może być kosztowne

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



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

$Diff(Red,Blue)$ $\quad Diff(Red,Green)$
$Diff(Green,Red)$ $\quad Diff(Green,Blue)$
$Diff(Blue,Red)$ $\quad Diff(Blue,Green)$

$Colorable()$ wynika wtw CSP ma rozwiązanie
Problemy CSP zawierają 3SAT jako szczególny przypadek, stąd dopasowywanie jest NP-trudne

Backward chaining: algorytm

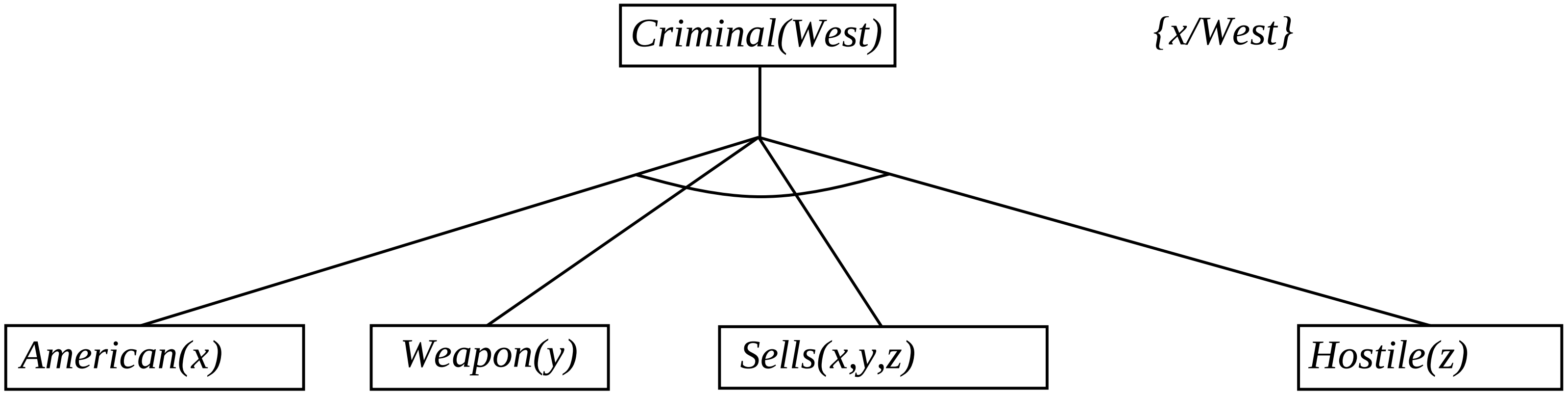
function FOL-BC-ASK(KB, goals, θ) **returns** a set of substitutions
inputs: KB, a knowledge base
$goals$, a list of conjuncts forming a query
θ, the current substitution, initially the empty substitution $\{\,\}$
local variables: ans, a set of substitutions, initially empty

if $goals$ is empty **then return** $\{\theta\}$
$q' \leftarrow$ SUBST(θ, FIRST($goals$))
for each r **in** KB where STANDARDIZE-APART(r) = ($p_1 \wedge \ldots \wedge p_n \Rightarrow q$)
and $\theta' \leftarrow$ UNIFY(q, q') succeeds
$ans \leftarrow$ FOL-BC-ASK(KB, $[p_1, \ldots, p_n | \text{REST}(goals)]$, COMPOSE($\theta'$, θ)) $\cup$ ans
return ans

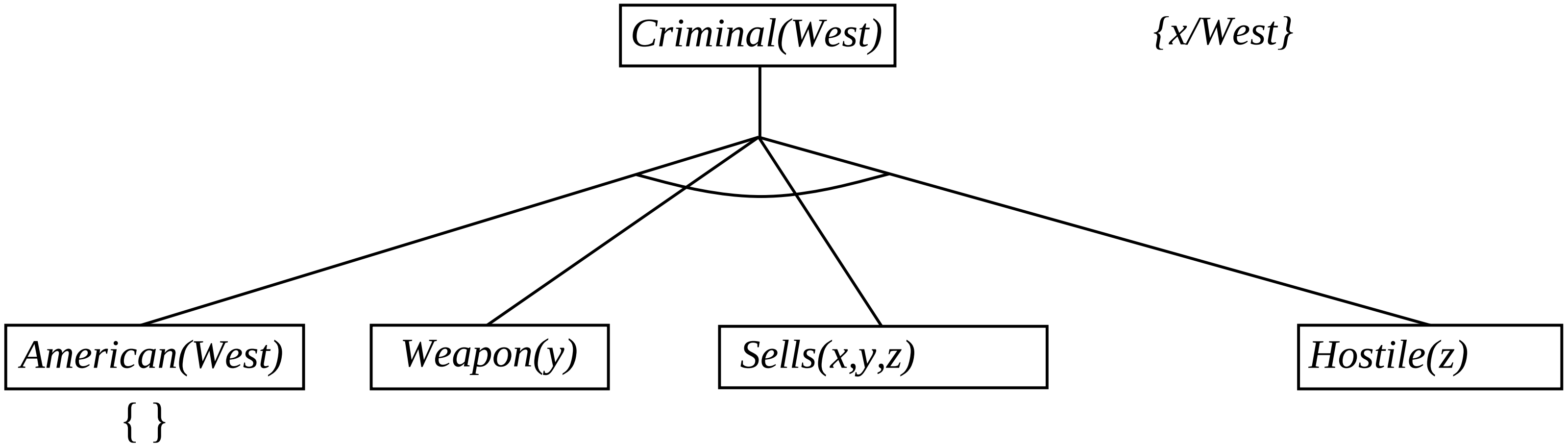
Backward chaining: przyklad

$\boxed{Criminal(West)}$

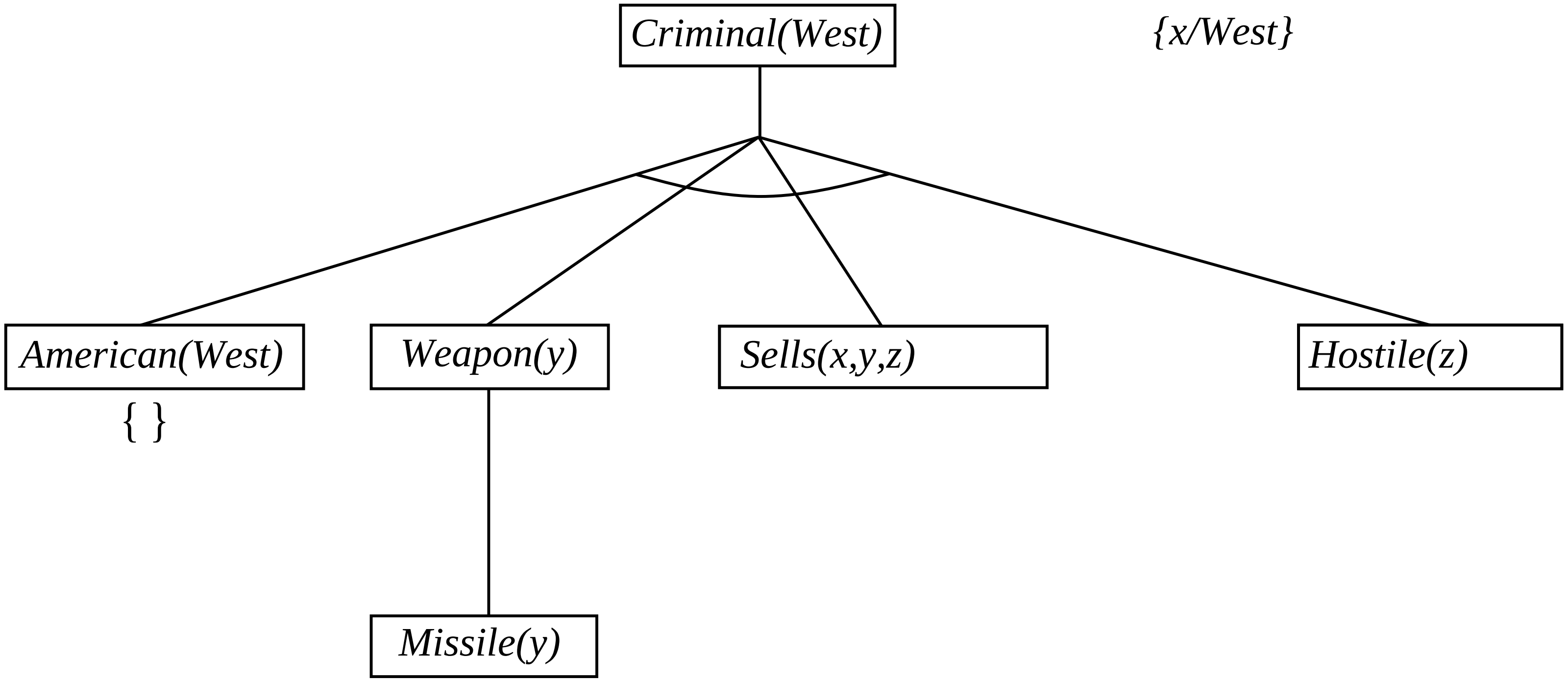
Backward chaining: przyklad



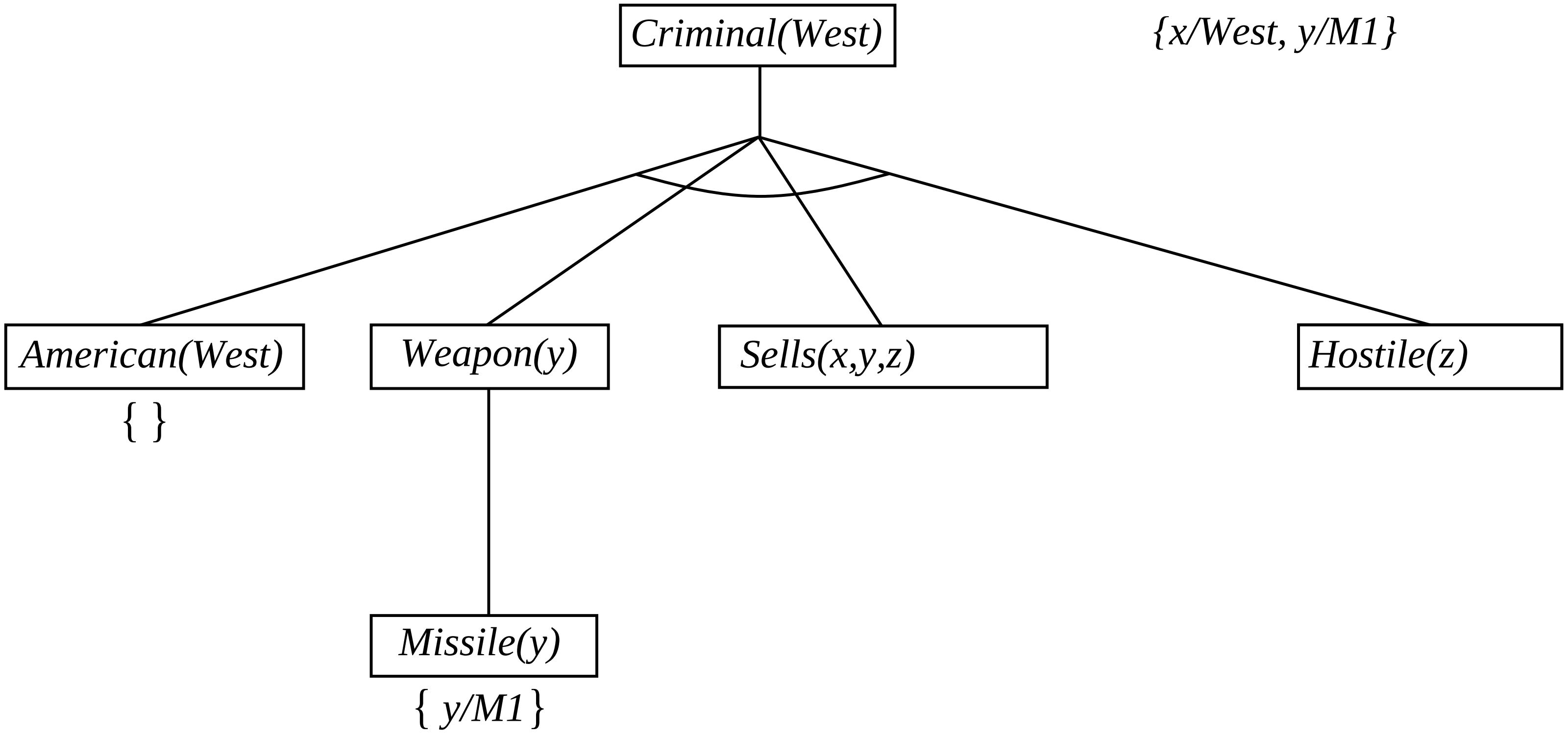
Backward chaining: przykład



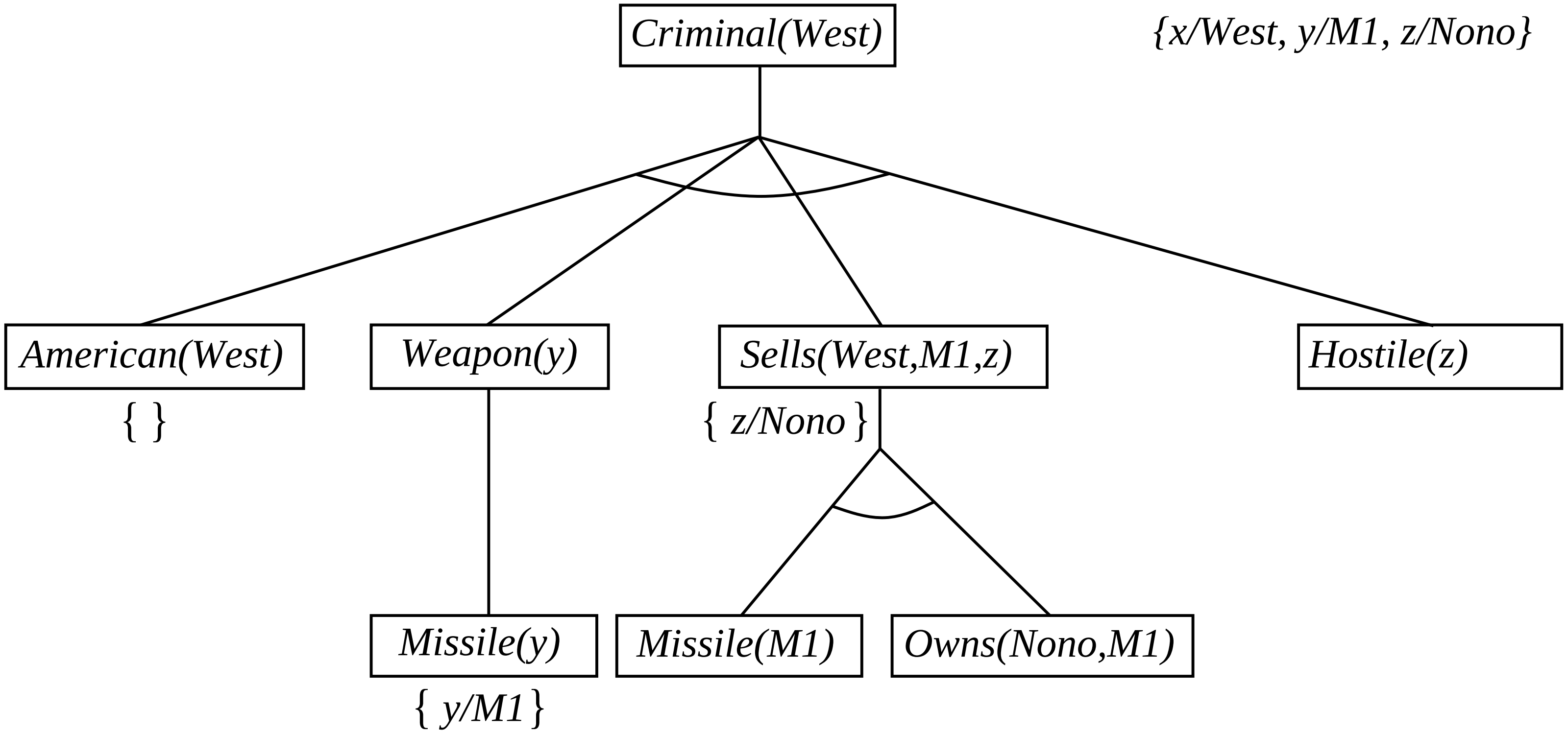
Backward chaining: przyklad



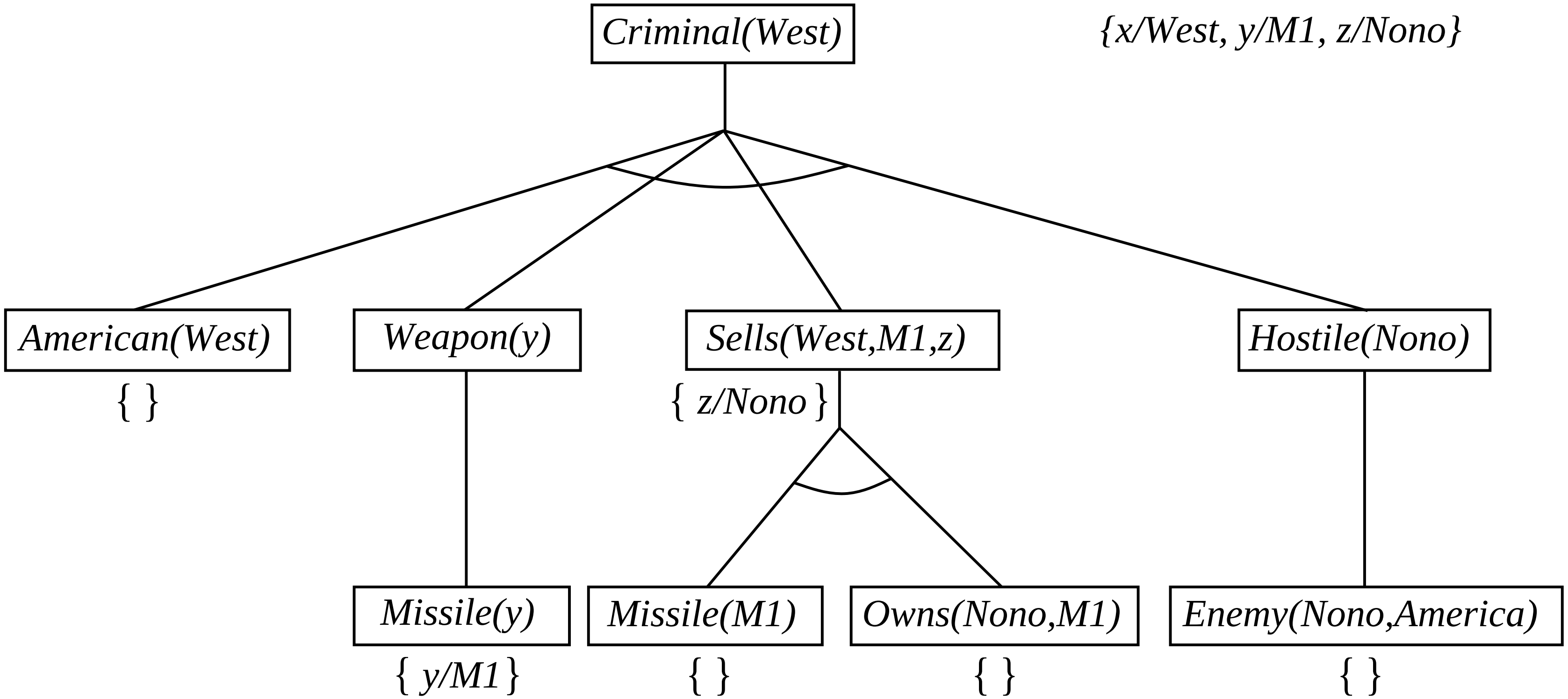
Backward chaining: przyklad



Backward chaining: przyklad



Backward chaining: przyklad



Backward chaining: wlasnosci

Przeszukiwanie rekurencyjne w głąb: pamięć jest liniowa od rozmiaru dowodu

Niepełny z powodu zapętleń
$\Rightarrow$ można poprawić sprawdzając każdy nowo wygenerowany cel,
czy nie wystąpił już wcześniej

Nieefektywny z powodu powtarzających się celów (udowodnionych i porażek)
$\Rightarrow$ można poprawić zapamiętując wcześniejsze wyniki
(wymaga dodatkowej pamięci!)

Powszechnie stosowany w programowaniu w logice (Prolog)

Rezolucja

Reguła rezolucji dla logiki I rzędu:

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m_1 \vee \cdots \vee m_n}{(\ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n)\theta}$$

gdzie $\textsc{Unify}(\ell_i, \neg m_j) = \theta$.

Na przykład,

$$\frac{\begin{array}{l}\neg Rich(x) \vee Unhappy(x) \\ Rich(Ken)\end{array}}{Unhappy(Ken)}$$

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):

$$\frac{\ell_1 \vee \cdots \vee \ell_i \vee \cdots \vee \ell_j \vee \cdots \vee \ell_k}{(\ell_1 \vee \cdots \vee \ell_i \vee \cdots \vee \ell_{j-1} \vee \ell_{j+1} \vee \cdots \vee \ell_k)\theta}$$

gdzie $\textsc{Unify}(\ell_i, \ell_j) = \theta$.

Na przykład,

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

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

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:

$$\forall\, x \;\; [\forall\, y \;\; Animal(y) \;\Rightarrow\; Loves(x,y)] \;\Rightarrow\; [\exists\, y \;\; Loves(y,x)]$$

1. Eliminacja równoważności i implikacji

$$\forall\, x \;\; [\neg\forall\, y \;\; \neg Animal(y) \lor Loves(x,y)] \lor [\exists\, y \;\; Loves(y,x)]$$

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

$$\forall\, x \;\; [\exists\, y \;\; \neg(\neg Animal(y) \lor Loves(x,y))] \lor [\exists\, y \;\; Loves(y,x)]$$
$$\forall\, x \;\; [\exists\, y \;\; \neg\neg Animal(y) \land \neg Loves(x,y)] \lor [\exists\, y \;\; Loves(y,x)]$$
$$\forall\, x \;\; [\exists\, y \;\; Animal(y) \land \neg Loves(x,y)] \lor [\exists\, y \;\; Loves(y,x)]$$

Rezolucja: przekształcanie do CNF

3. Przemianowanie zmiennych tak, żeby każdy kwaantyfikator miał inną

$$\forall x \;\; [\exists y \;\; Animal(y) \land \neg Loves(x,y)] \lor [\exists z \;\; Loves(z,x)]$$

4. Skolemizacja: bardziej ogólna postać egzystencjalnej instancjacji.
   Każda zmienna kwantyfikowana egzystencjalnie jest zastępowana przez funkcję Skolema otaczających ją uniwersalnie kwantyf. zmiennych:

$$\forall x \;\; [Animal(F(x)) \land \neg Loves(x, F(x))] \lor Loves(G(x), x)$$

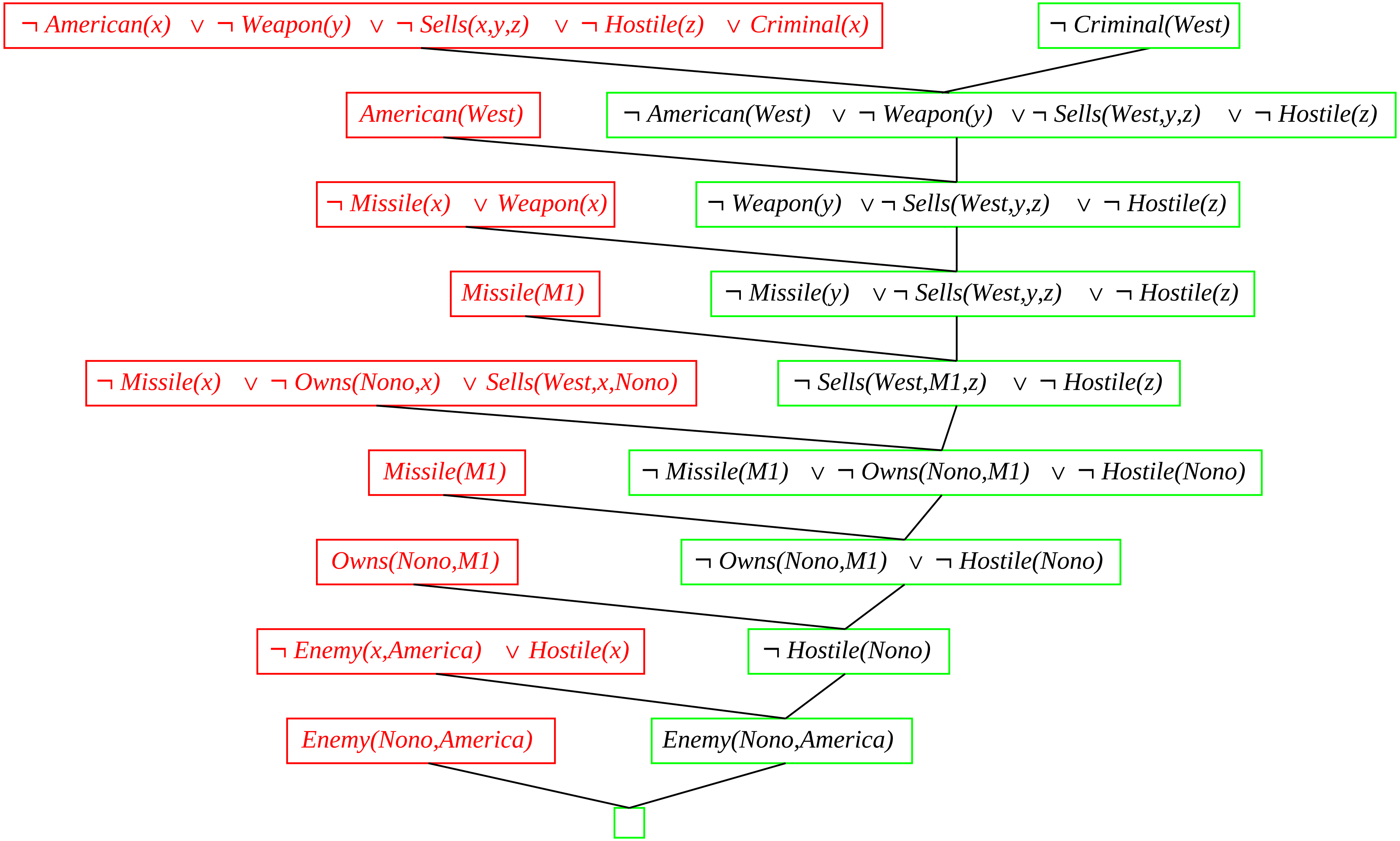
5. Usunięcie kwantyfikatorów uniwersalnych:

$$[Animal(F(x)) \land \neg Loves(x, F(x))] \lor Loves(G(x), x)$$

6. Rozdzielenie $\land$ względem $\lor$:

$$[Animal(F(x)) \lor Loves(G(x), x)] \land [\neg Loves(x, F(x)) \lor Loves(G(x), x)]$$

Rezolucja: przyklad



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 $\subset [0, 1]$