

# Interpolačná lemma

## *zopakovanie*

### **Vlastnosti jednoduchého dôsledku:**

- Každý podgraf grafu je jeho jednoduchým dôsledkom.
- Graf je jednoduchým dôsledkom každej svojej inštancie.

### **Interpolation lemma:**

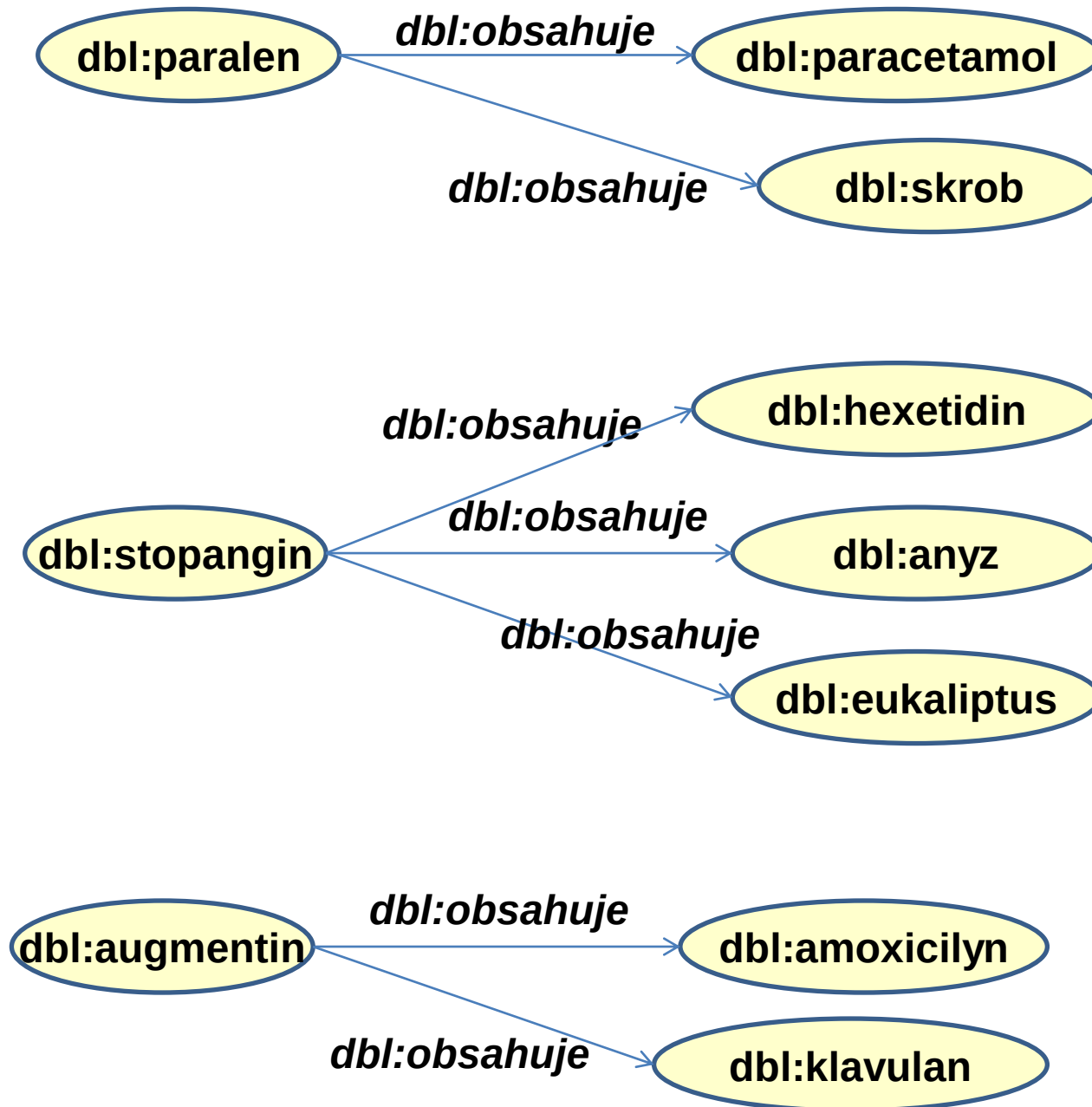
Graf E je jednoduchým dôsledkom grafu G práve vtedy, keď existuje inštancia grafu E, ktorá je podgrafom G.

**Dôkaz:** nájdete v <http://www.w3.org/TR/rdf11-mt/#proofs-of-some-results-informative>

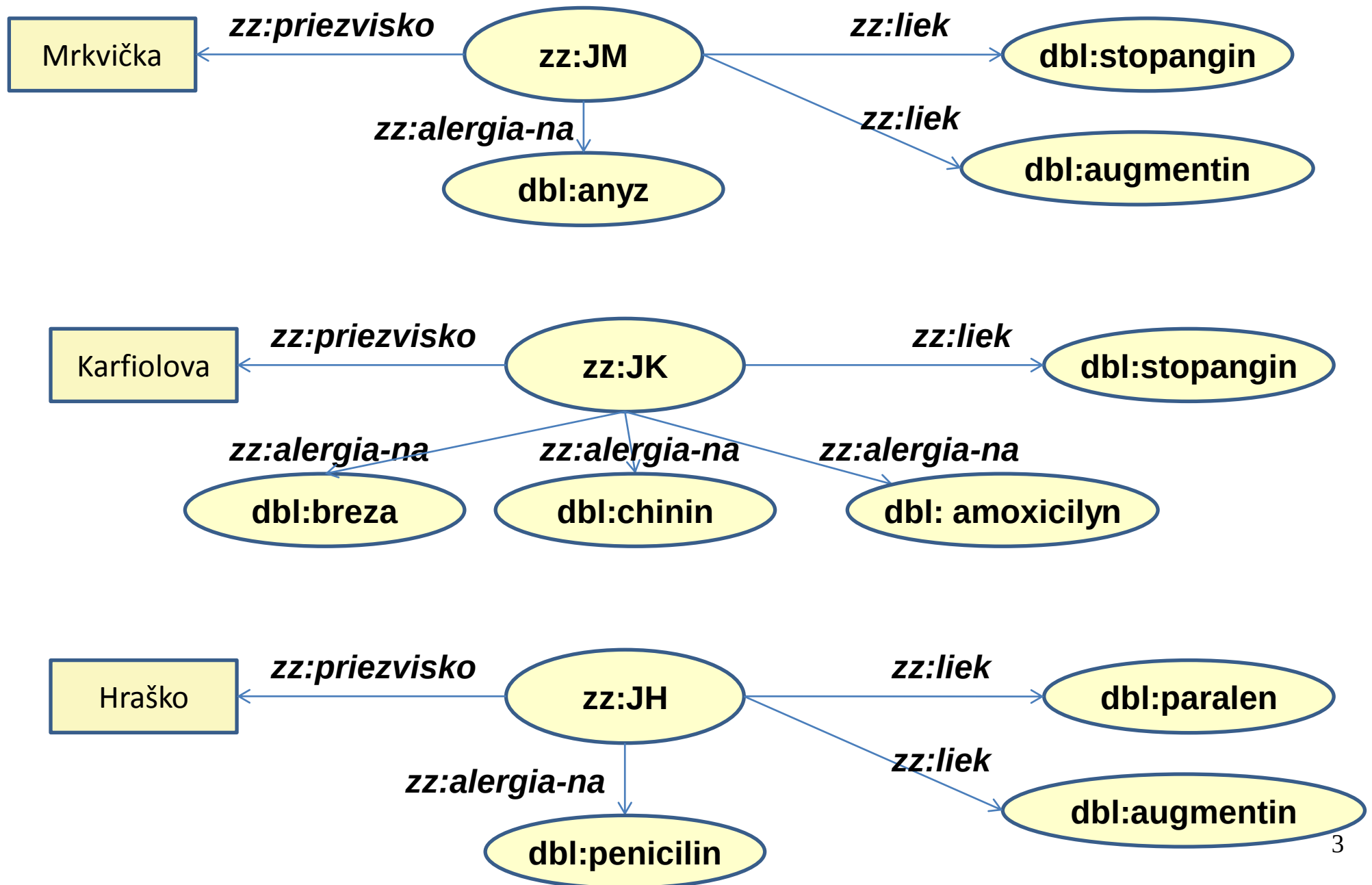
*Táto lemma kompletne charakterizuje jednoduchý dôsledok len pomocou syntaktických pojmov.*

*Zároveň dáva návod ako zistiť či jeden graf vyplýva z druhého.*

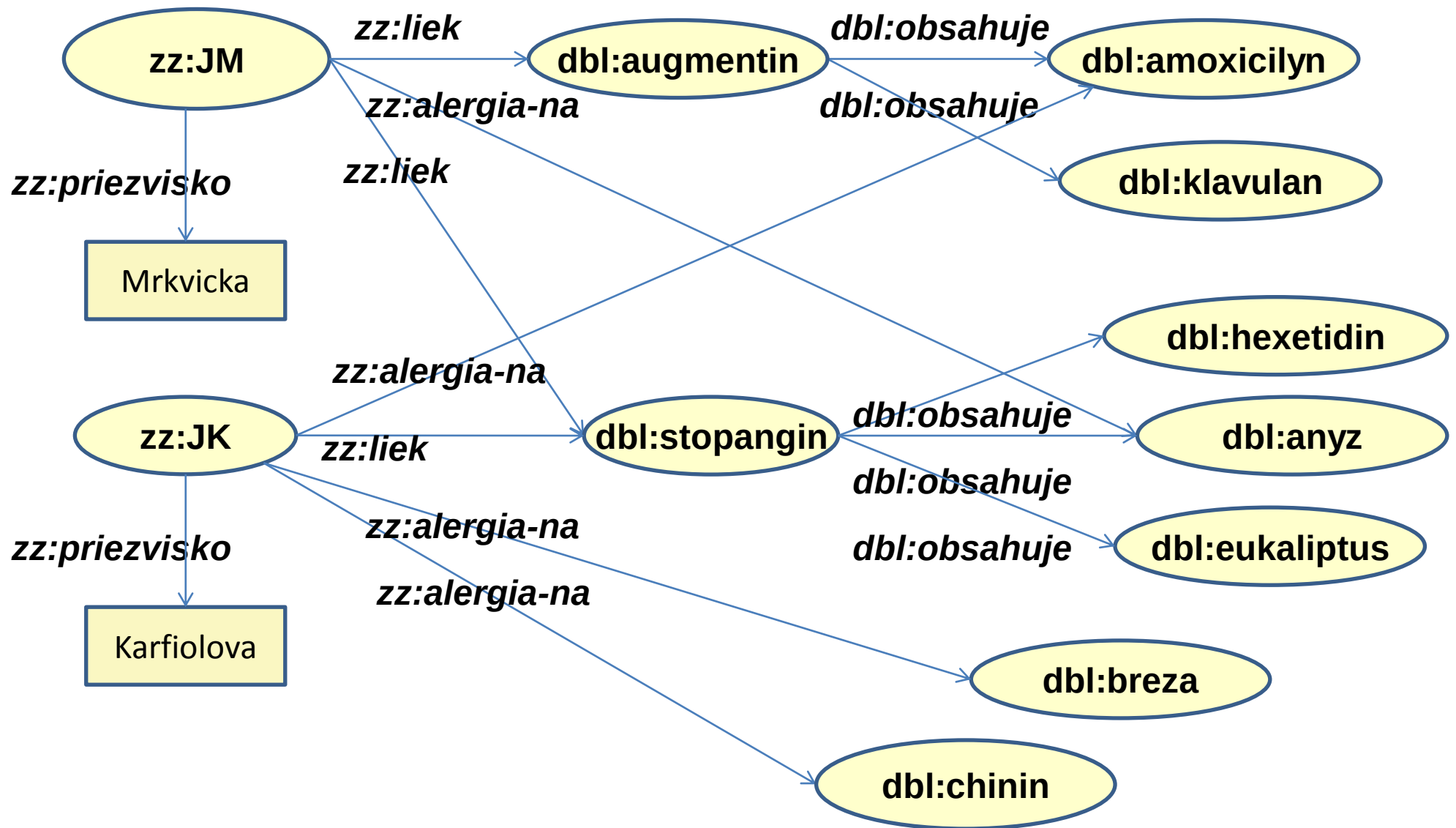
# Príklad - databáza liekov



# Príklad - zdravotné záznamy



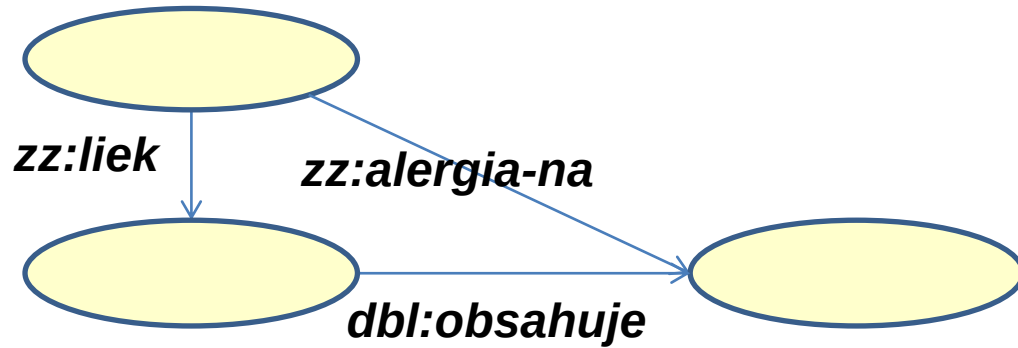
# Príklad - zlúčené databázy



nema niektorý pacient predpisany liek, ktorý by mohol spôsobiť alergickú reakciu? <sup>4</sup>

# Príklad - hypotéza

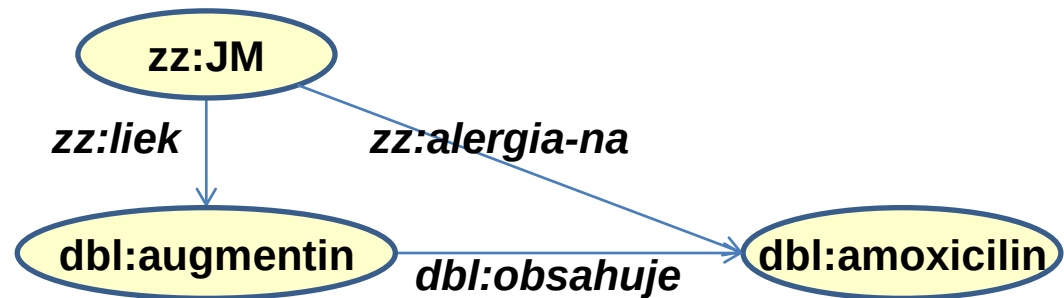
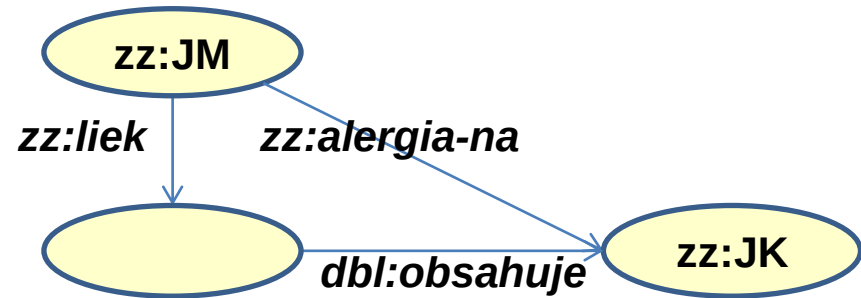
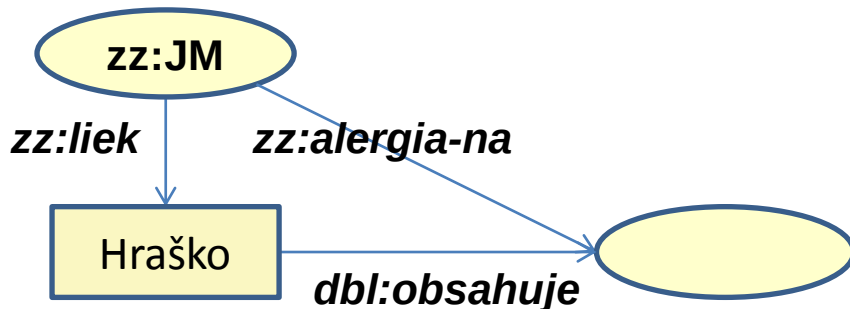
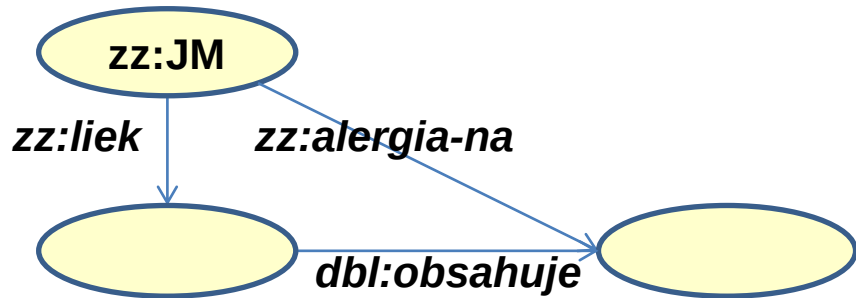
## Graf hypotézy H:



## Algoritmus:

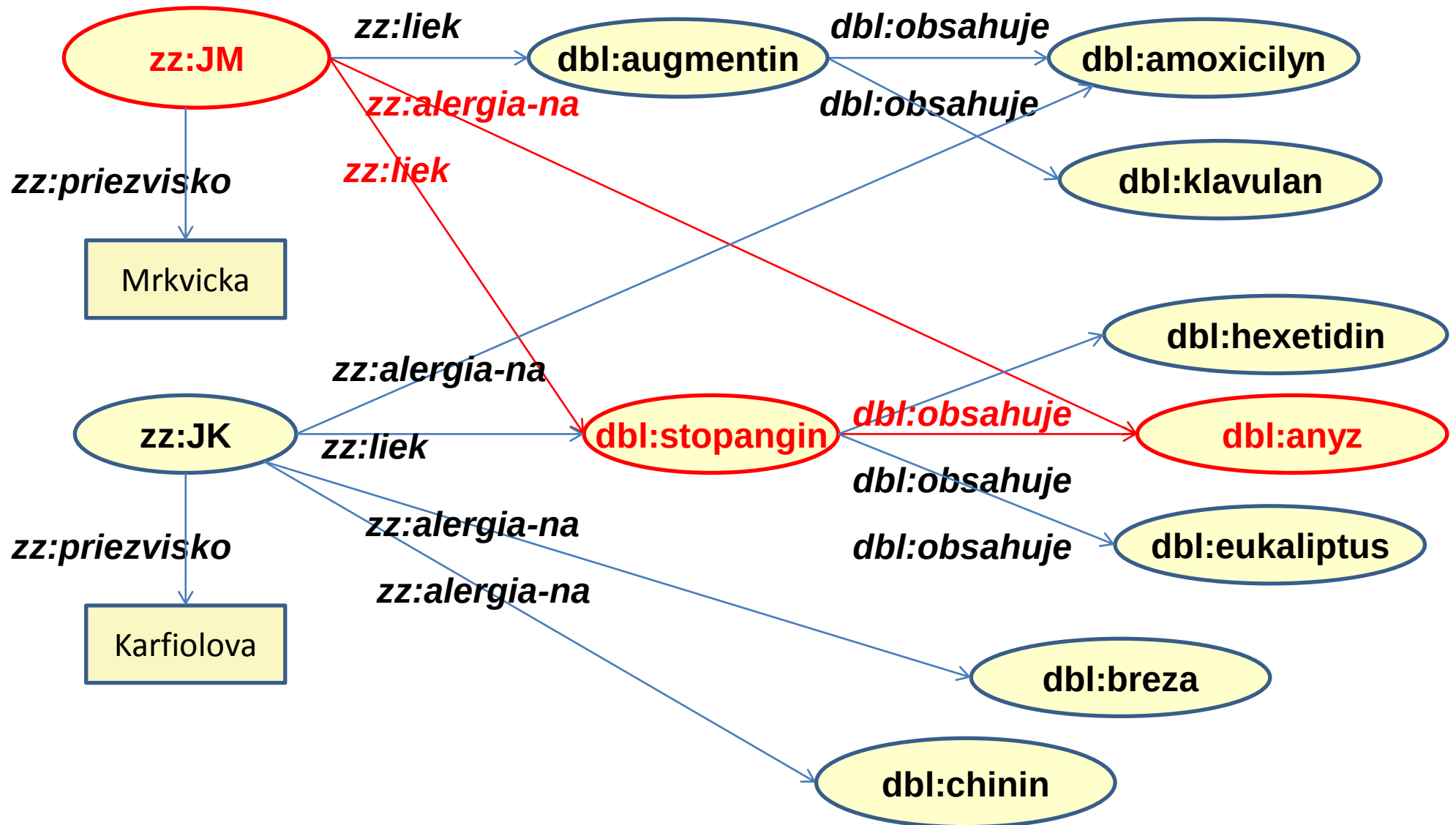
```
for each K in instancia(H):  
    if K je_podgraf G:  
        return True  
return False.
```

## Inštancie grafu H:



žiadna z týchto nie je podgrafom G, ale tomto prípade jedna naozaj existuje...

# Match



# Izomorfizmus a logická ekvivalencia grafov

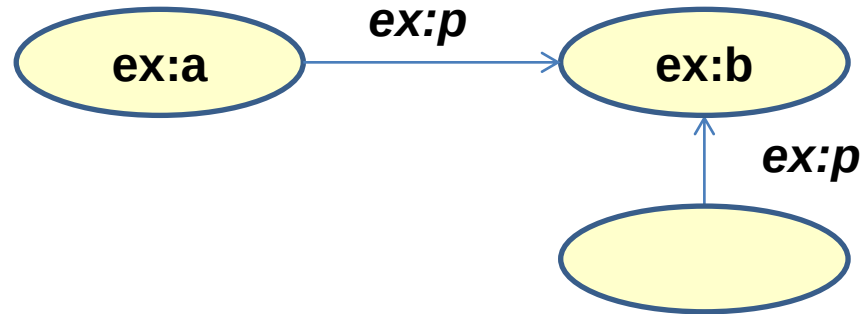
## *ešte raz*

Nech grafy **G** a **F** sú **izomorfné**, teda sú si navzájom inštanciami. Z interpolačnej lemy vyplýva, že sú potom aj **logicky jednoducho ekvivalentné**.

Ako ukazuje nasledujúci príklad, opačné tvrdenie platiť nemusí:

**F**

ex:a ex:p ex:b .  
\_:y ex:p ex:b .



**G**

ex:a ex:p ex:b .



G a F su logicky ekvivalentné, ale nie izomorfné (ukážte) .

**F obsahuje svoju inštanciu - Graf G, ako podgraf**, teda ho z logického hladiska môžeme zredukovať na tento podgraf a ostatne trojice vypustiť – sú redundantné.

# Štíhle (lean) grafy

Ak **graf obsahuje svoju inštanciu ako podgraf**, môžeme ho z logického hľadiska zredukovať na tento podgraf, a ostatné trojice z neho vypustiť – sú redundantné.

**Definícia:** **Štíhly graf** (lean) je graf, ktorý neobsahuje žiadnu svoju inštanciu ako vlastný podgraf.

**Tvrdenie:** Dva štíhle grafy sú jednoducho logicky ekvivalentné práve vtedy, keď sú izomorfné.

**Príklad:**

ex:a ex:p \_:x .  
\_:y ex:p \_:x .

nie je lean graf. Ktoré tvrdenie možno vynechať? Zdôvodnite

**Príklad**

Je nasledujúci graf tiež non-lean?

ex:a ex:p \_:x .  
\_:x ex:p \_:x .



# Algoritmus jednoduchéj dedukcie

*Ak báza znalostí, obsahuje trojicu reprezentujucu informáciu, ktorá však logicky vyplýva z ostatných - je ich dôsledkom, potom môžeme túto trojicu z bázy znalostí vypustiť – logický význam KB sa nezmení.*

*Plati to však aj naopak: **Implicitné** dôsledky KB, t.j. fakty, ktoré síce nie sú **explicitne** vyjadrené v KB ale z nej logicky vyplývajú, môžeme do KB pridať.*

*To nám umožňuje navrhnúť modifikáciu algoritmu (overovania dôsledku) založenú na **dedukcii**: Namiesto toho aby sme generovali instance H a overovali či nie sú podgrafom G, môžeme do grafu G pridávať jeho logické dôsledky a skúšať či H nie je podgrafom rozšíreného grafu.*

## Algoritmus jednoduchéj dedukcie:

Nech D je zoznam všetkých trojíc, ktoré sú jednoduchým dôsledkom G  
ale nie sú v G.

do:

if K je\_podgraf G:

return True

else

vyber trojicu z D a pridaj ju do G

return False.

*ako však generovať zoznam D?*

# Pravidlo jednoduchéj dedukcie

Čo sú dôsledky v jednoduchéj interpretácii? Pripomeňme si:

1. Každý podgraf grafu je jeho jednoduchým dôsledkom – **podgraf netreba pridávať**
2. Graf je jednoduchým dôsledkom každej svojej inštancie.

Druhé tvrdenie nám dáva jednoduchý návod ako generovať užitočné dôsledky:

Ak v trojici nahradíme subjekt alebo objekt novým prázdny uzlom dostaneme novú trojicu, ktorá je jednoduchým dôsledkom pôvodnej trojice. Pôvodna trojica je totiž jej inštanciou.

## Pravidlo jednoduchéj dedukcie

- pre objekt:

$ex:s \quad ex:p \quad ex:o.$

---

$ex:s \quad ex:p \quad \_ :x$

Vyjadrené v jazyku predikátovej logiky pravidlo hovorí:  $p(s,o) \Rightarrow (\exists x) p(s,x)$

- pre subjekt:

$ex:s \quad ex:p \quad ex:o.$

---

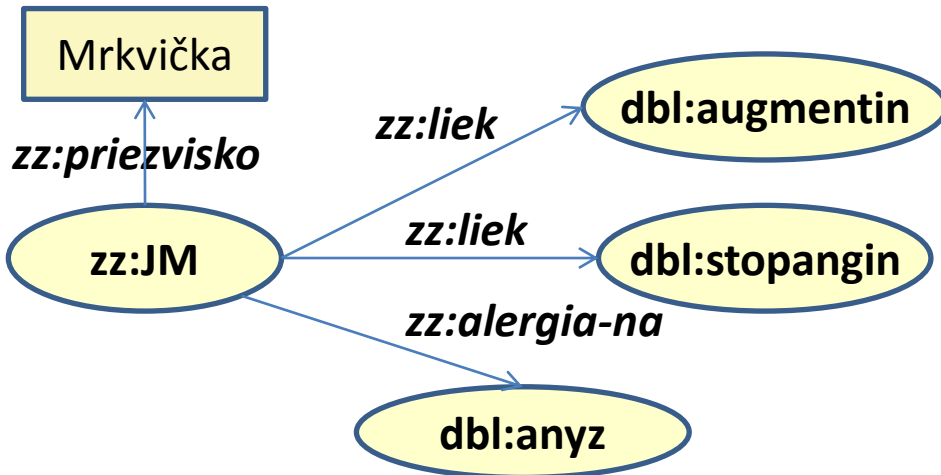
$\_ :x \quad ex:p \quad ex:o$

vyjadrené v jazyku predikátovej logiky:  $p(s,o) \Rightarrow (\exists x) p(x,o)$

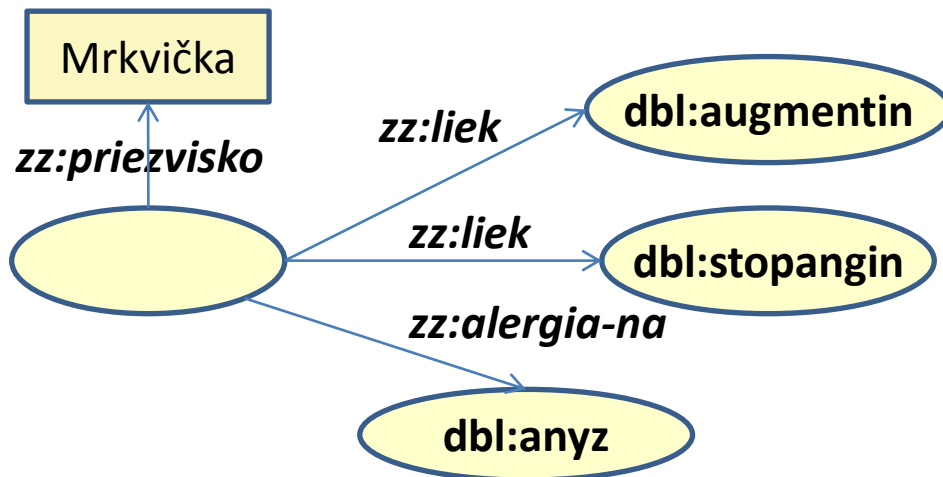
# Algoritmus jednoduchéj dedukcie

To isté môžeme urobiť s každým uzlom grafu:

Vezmeme podgraf obsahujúci všetky hrany v ktorých sa uzol nachádza, vytvoríme nový prázdny uzol a nahradíme ním pôvodný uzol vo všetkých trojiciach podgrafu. Výsledný podgraf bude jednoduchým dôsledkom pôvodného grafu.



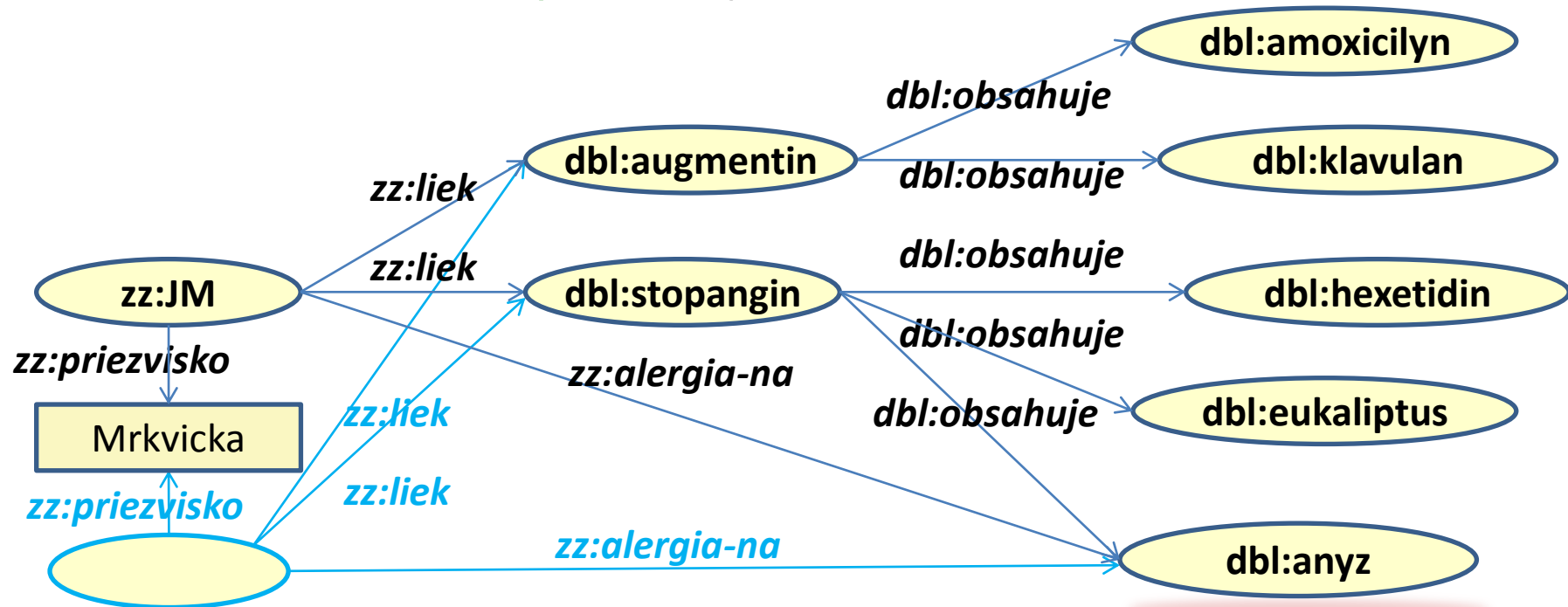
**zz:JM**    **zz:priezvisko**    **"Mrkvička";**  
**zz:liek**                    **dbl:augmentin;**  
**zz:liek**                    **dbl:stopangin;**  
**zz:alergia-na**           **dbl:anyz.**



**\_ :x1**    **zz:priezvisko**    **"Mrkvička";**  
**zz:liek**                    **dbl:augmentin;**  
**zz:liek**                    **dbl:stopangin;**  
**zz:alergia-na**           **dbl:anyz.**

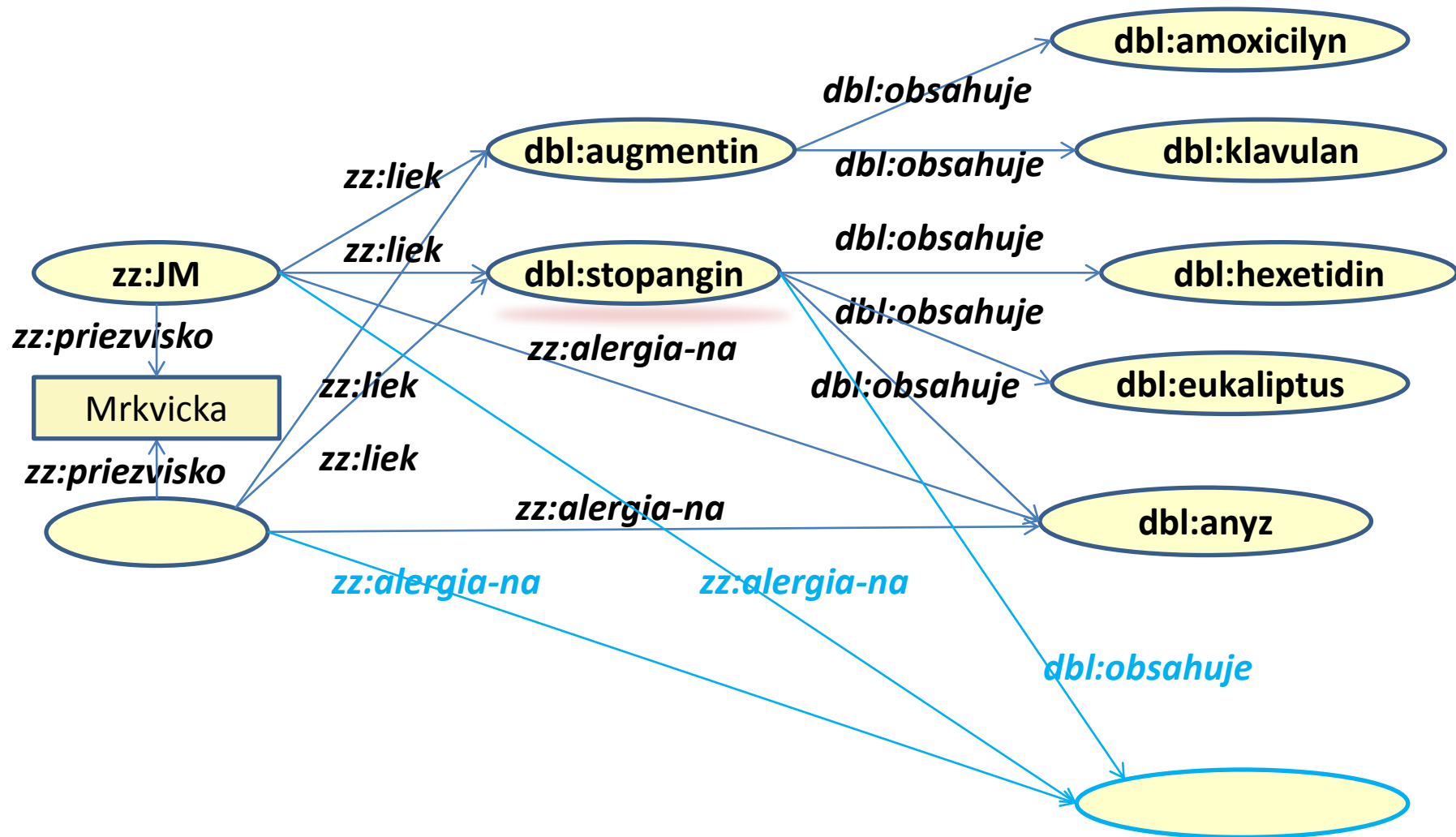
# Algoritmus jednoduché dedukcie – príklad

Takto vygenerované trojice – dôsledky môžeme pridať do grafu a to isté následne urobiť s ďalším uzlom - napr. `dbl:anyz`



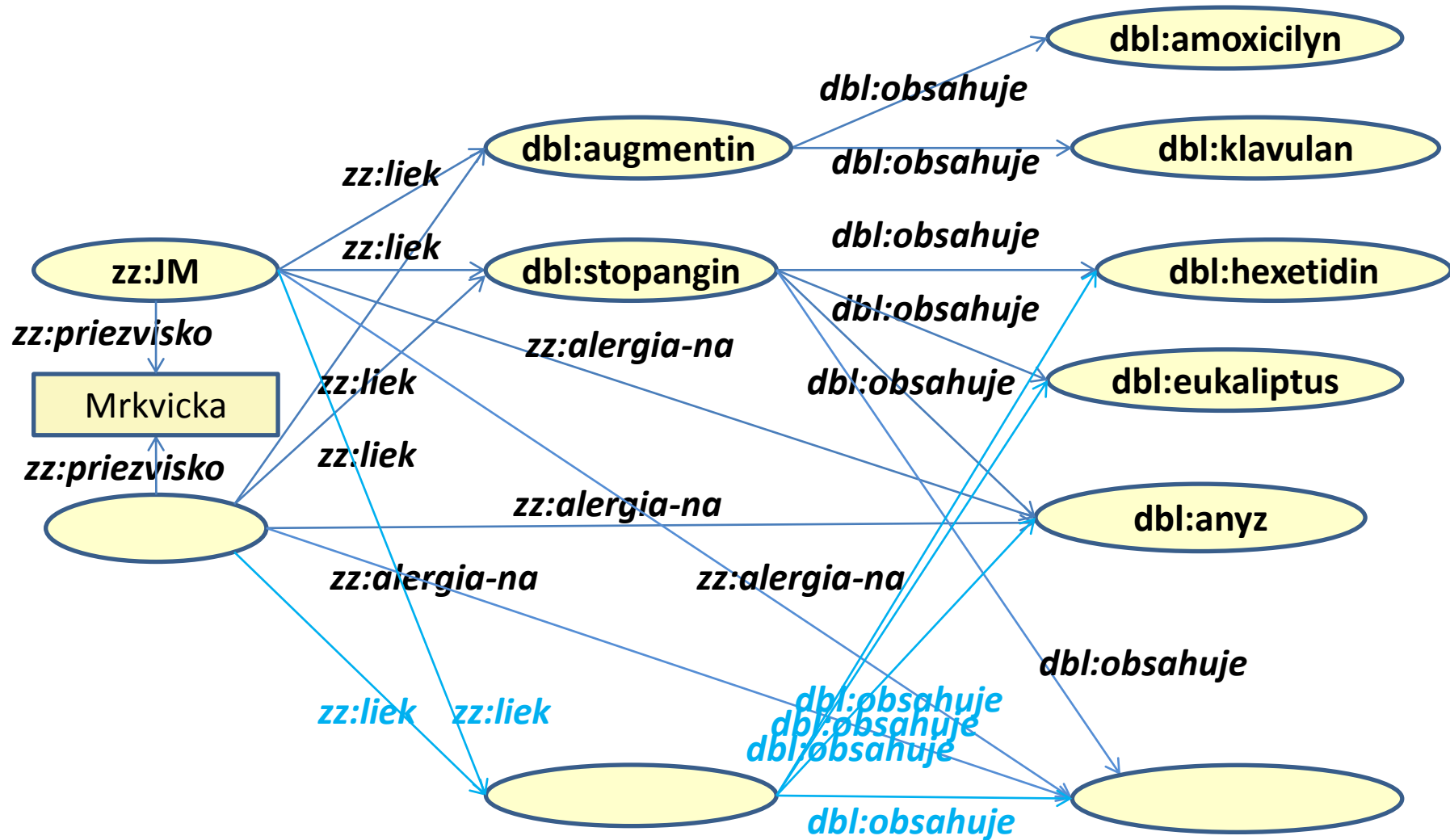
# Algoritmus jednoduché dedukcie – príklad

A teraz s uzlom dbl:stopangin



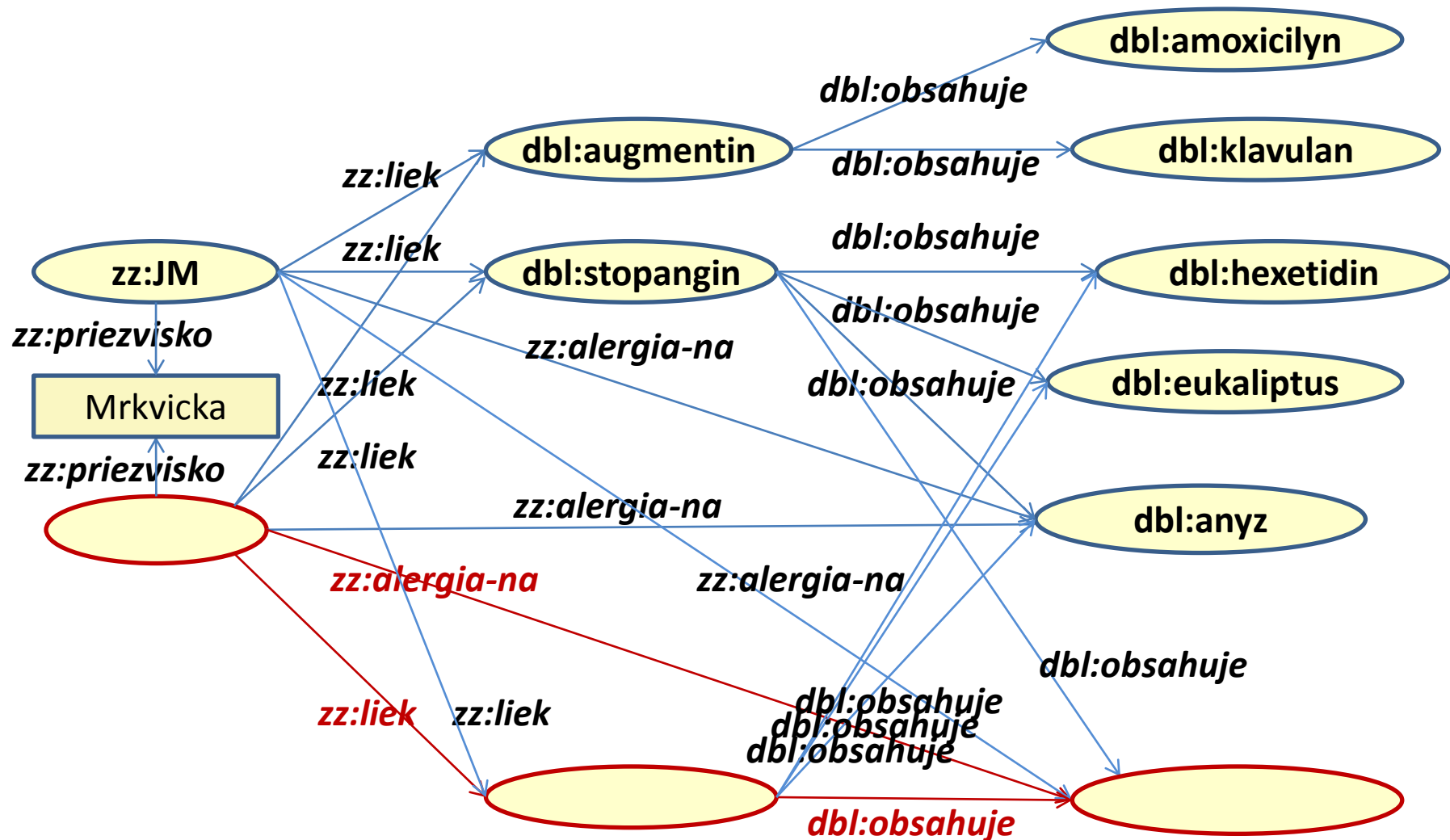
# Algoritmus jednoduché dedukcie – príklad

A teraz s uzlom dbl:stopangin



# Algoritmus jednoduchéj dedukcie – príklad

A máme hľadaný dôsledok



# Algoritmus jednoduchéj dedukcie

Pravidlo jednoduchéj dedukcie dáva návod na **korektný algoritmus**, ktorý sa pokúša overiť či jeden graf je logickým dôsledkom iného grafu v jednoduchéj interpretácii.

## Algoritmus jednoduchéj dedukcie:

- Vychádzajúc z grafu  $G$  postupne ho rozširujeme o trojice na základe jednoduchého dedukčného pravidla.
  - Ak rozšírený graf obsahuje graf  $H$  ako podgraf, potom algoritmus skončí konštatovaním  $G \models H$ .
  - Ak už nie je možné ďalej pridať použitím pravidla nové trojice do  $G$  a  $H$  algoritmus skončí.

**Veta:** Graf  $H$  je jednoduchým logickým dôsledkom grafu  $G$ , ak  $G$  môže byť rozšírený aplikovaním jednoduchého dedukčného pravidla do grafu  $G'$ , ktorý obsahuje  $H$  ako podgraf.

**Dôsledok:** Algoritmus jednoduchéj dedukcie je **korektná a úplná** procedúra pre určenie či jeden graf je jednoduchým logickým dôsledkom iného grafu.



# Ďalšie dedukčné pravidlá a rôzne triedy interpretácií

## Spracované podľa

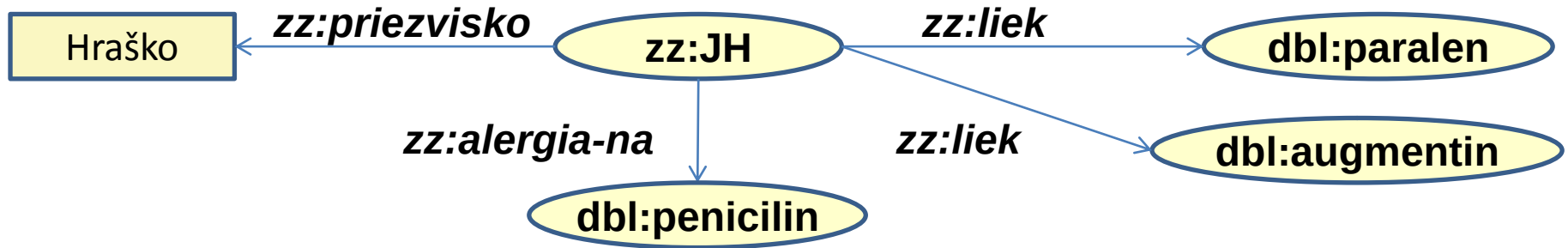
dokumentov W3C

- <http://www.w3.org/TR/rdf11-mt/>

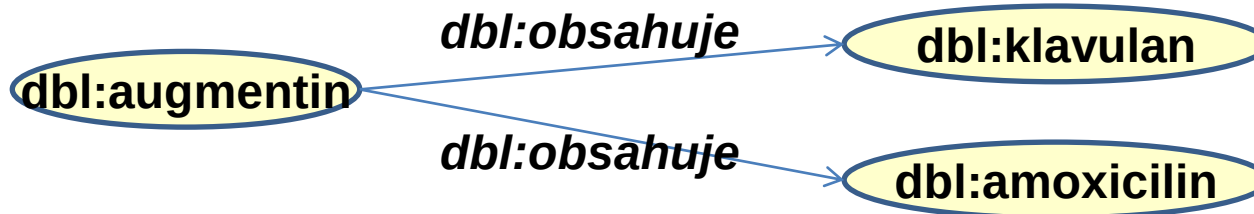
a prednášok

- <http://www.inf.tu-dresden.de/content/institutes/ki/cl/study/summer14/fswt/slides/FSWT2014-L4-RDSF-Semantics.pdf>

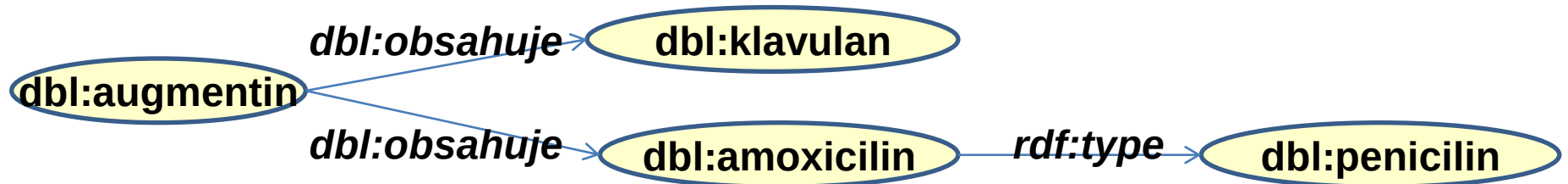
# Pokračovanie príkladu



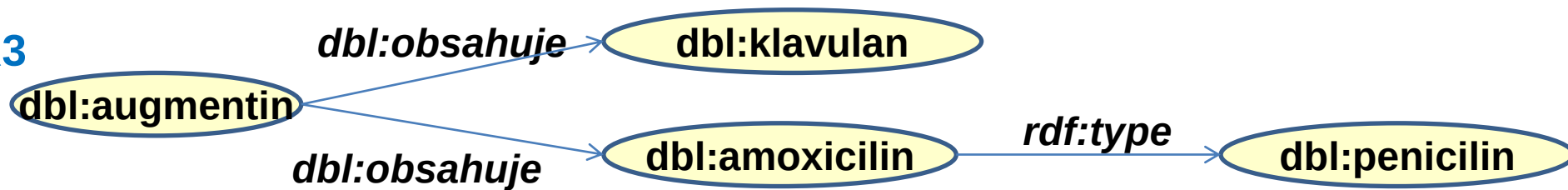
## Databaza liekov GR2



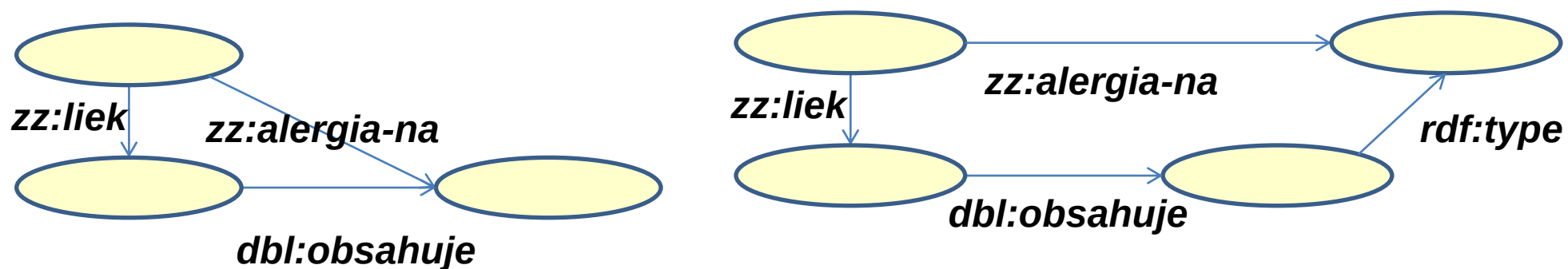
## GR3



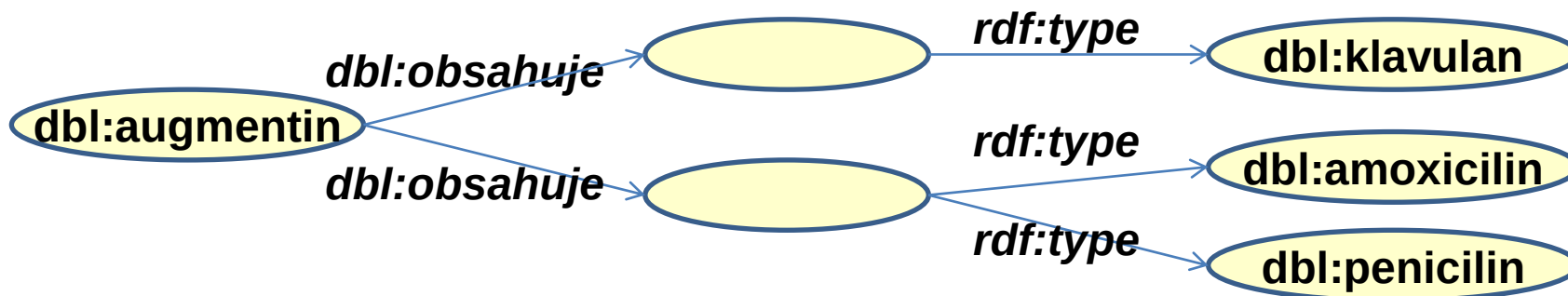
GR3



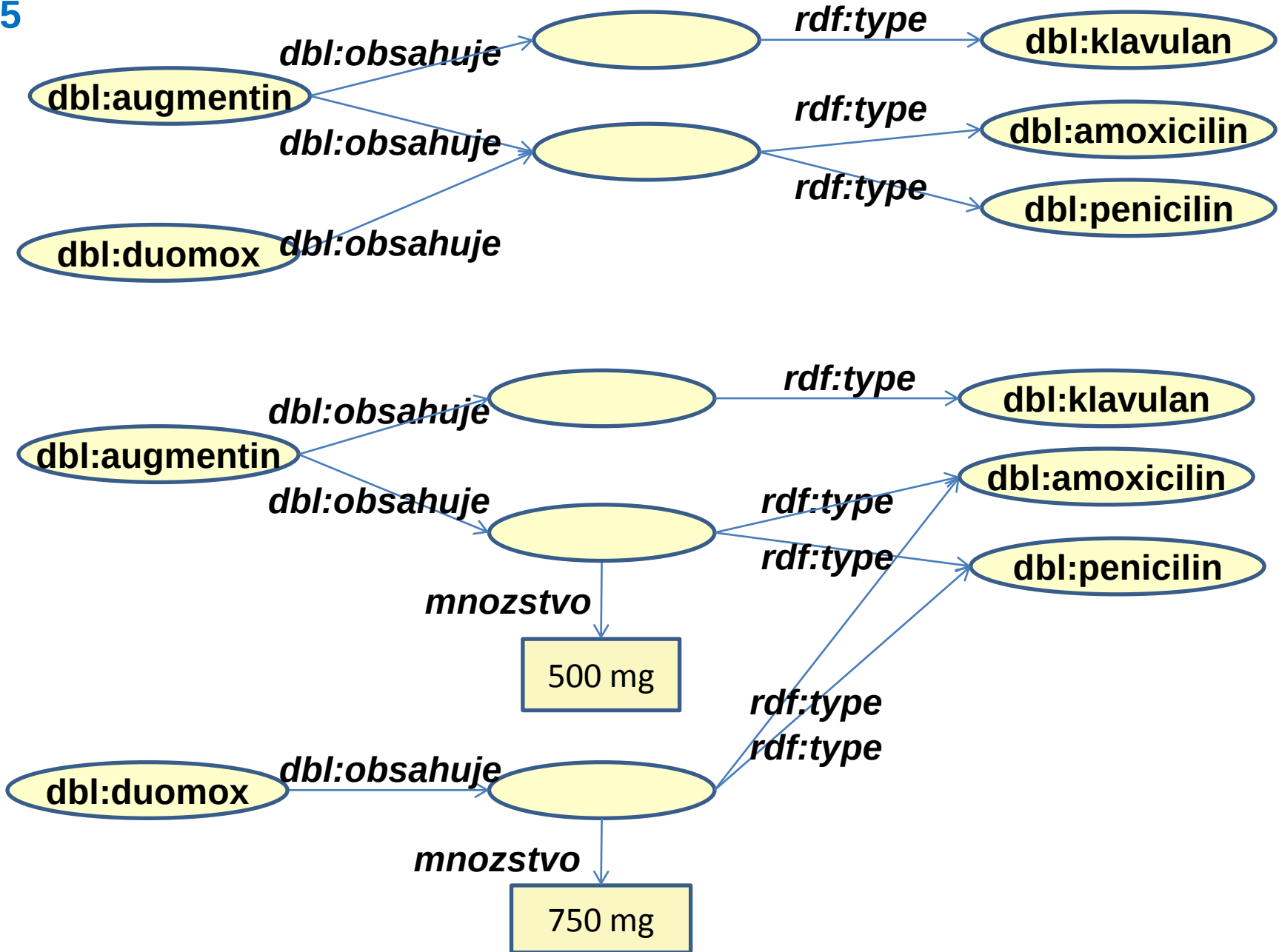
H3

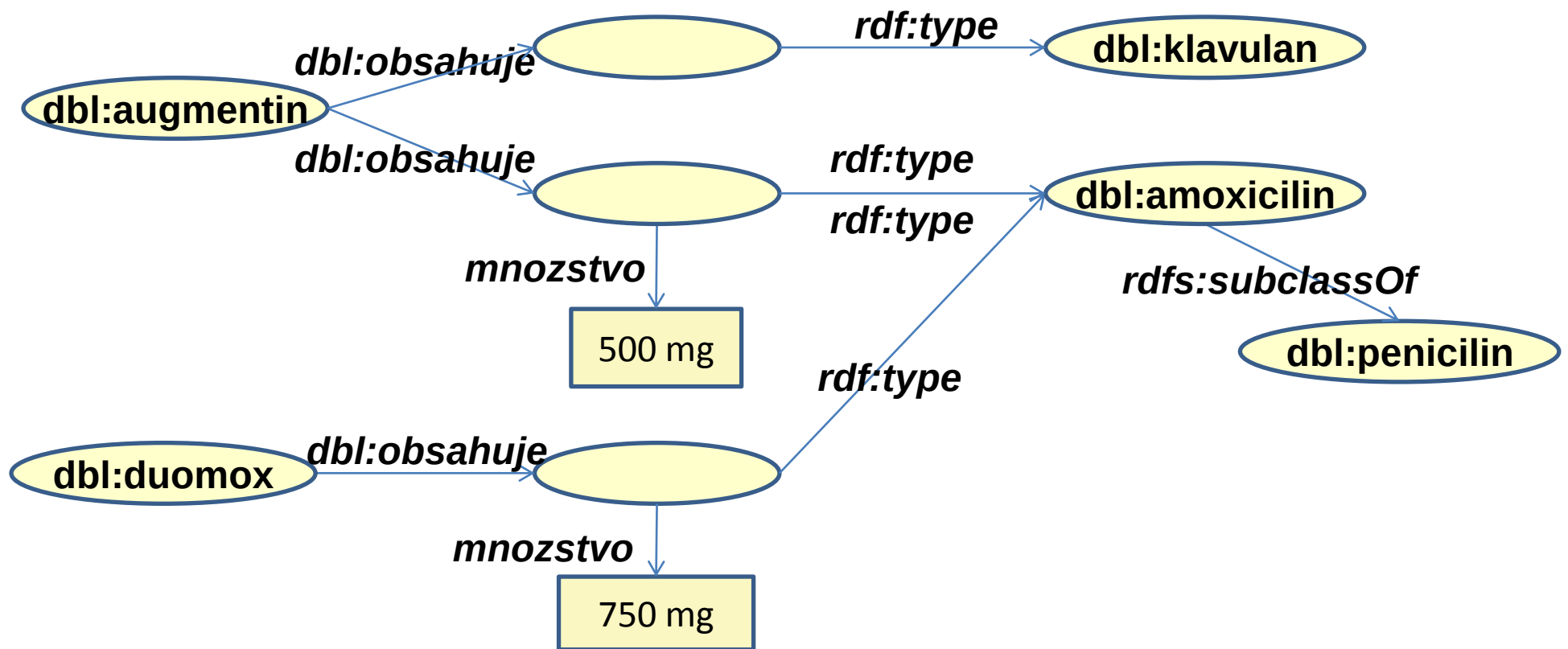
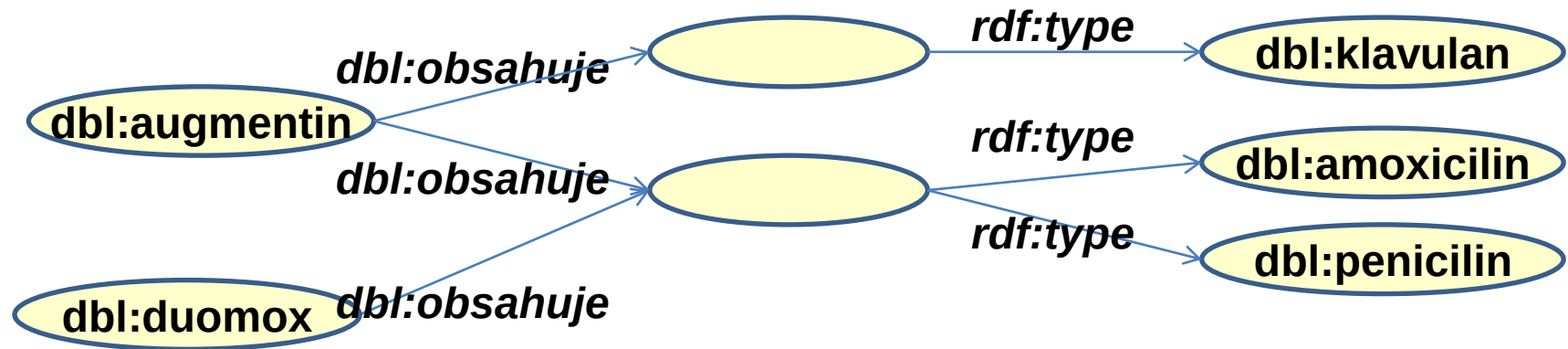


GR4

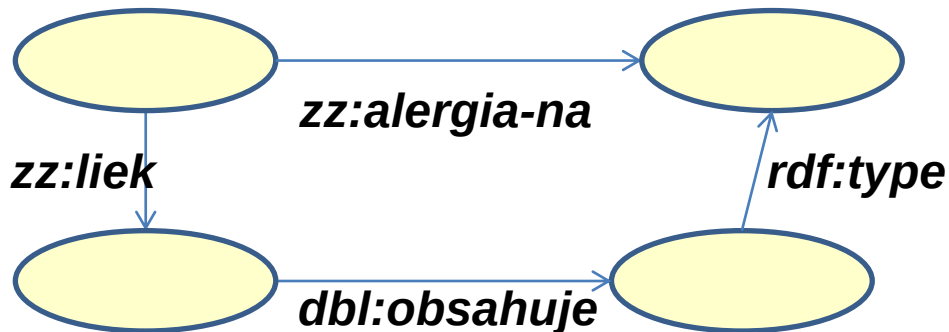
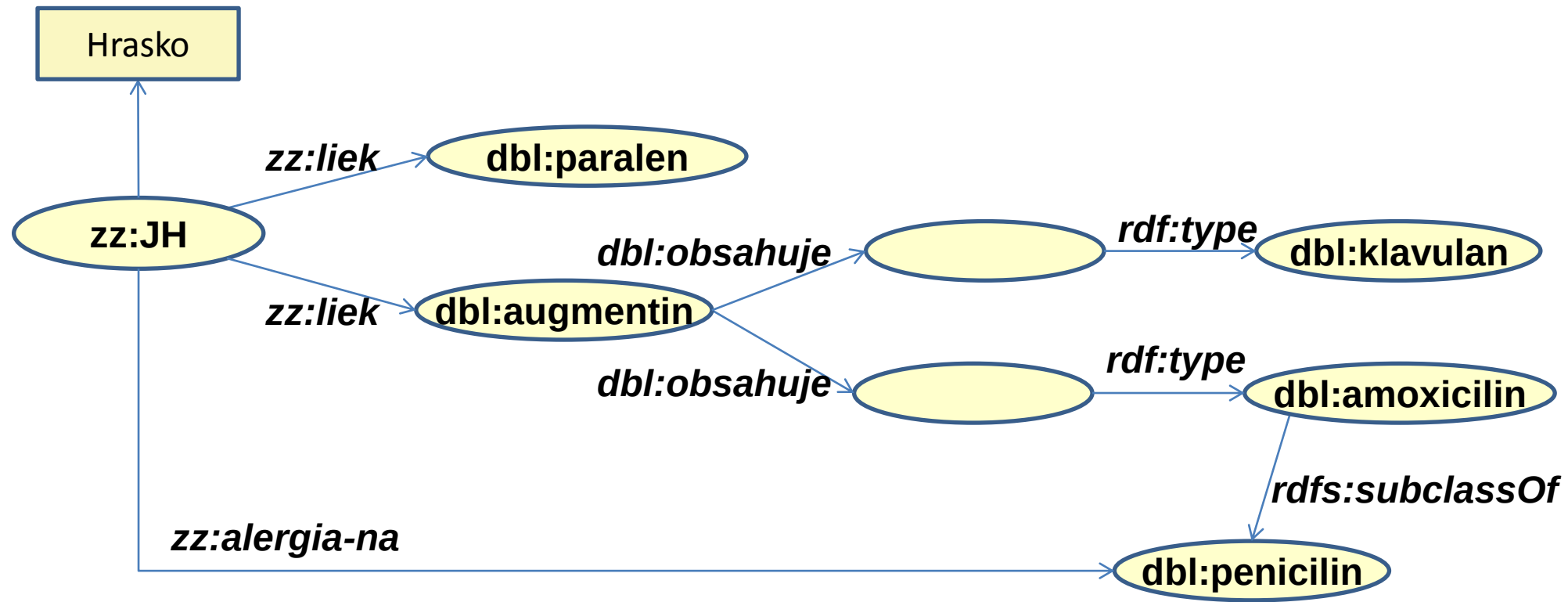


GR5



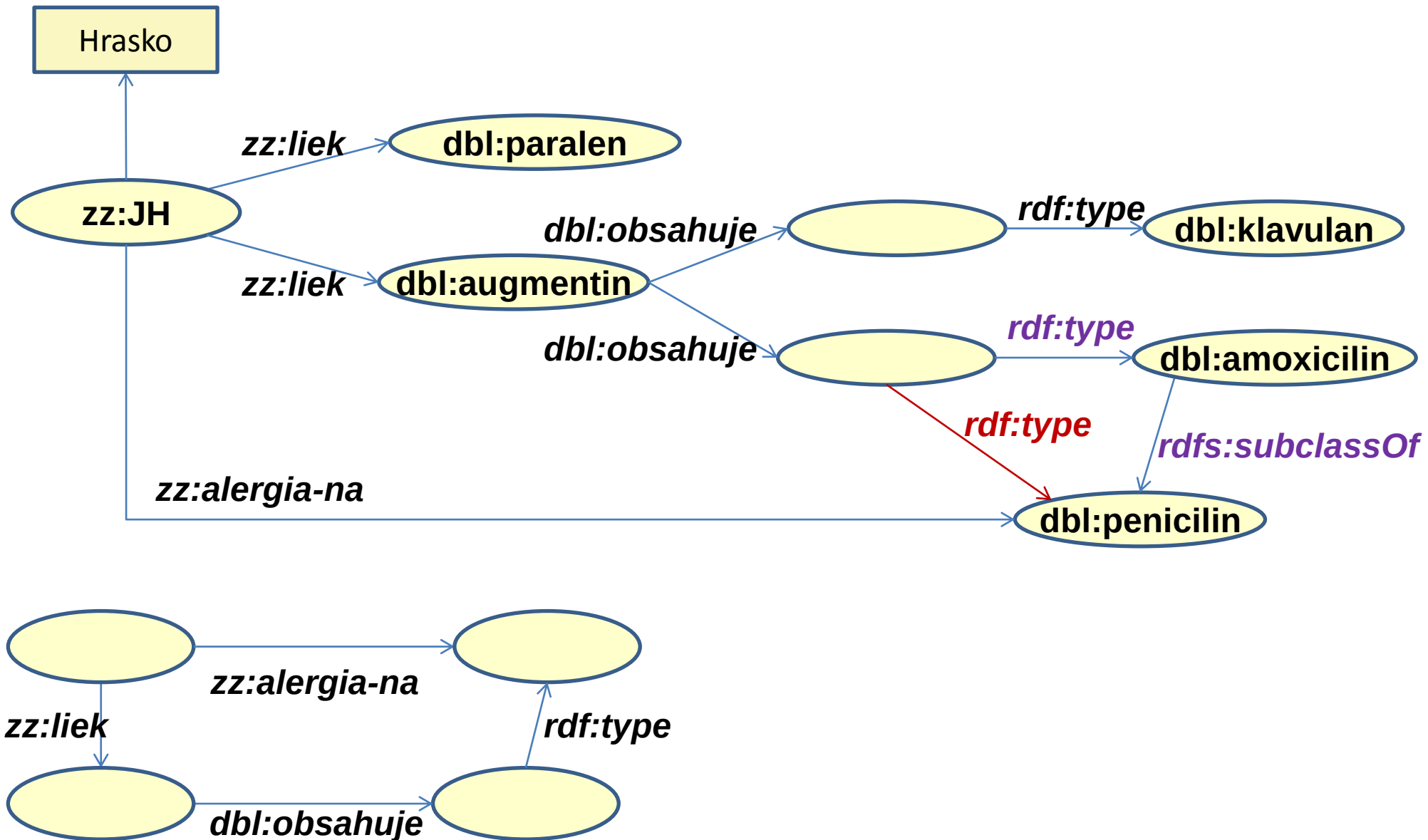


# Dedukčné pravidlo pre rdf:type

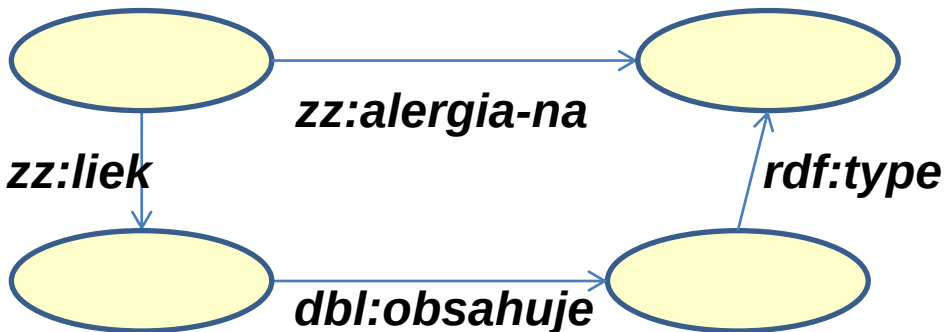
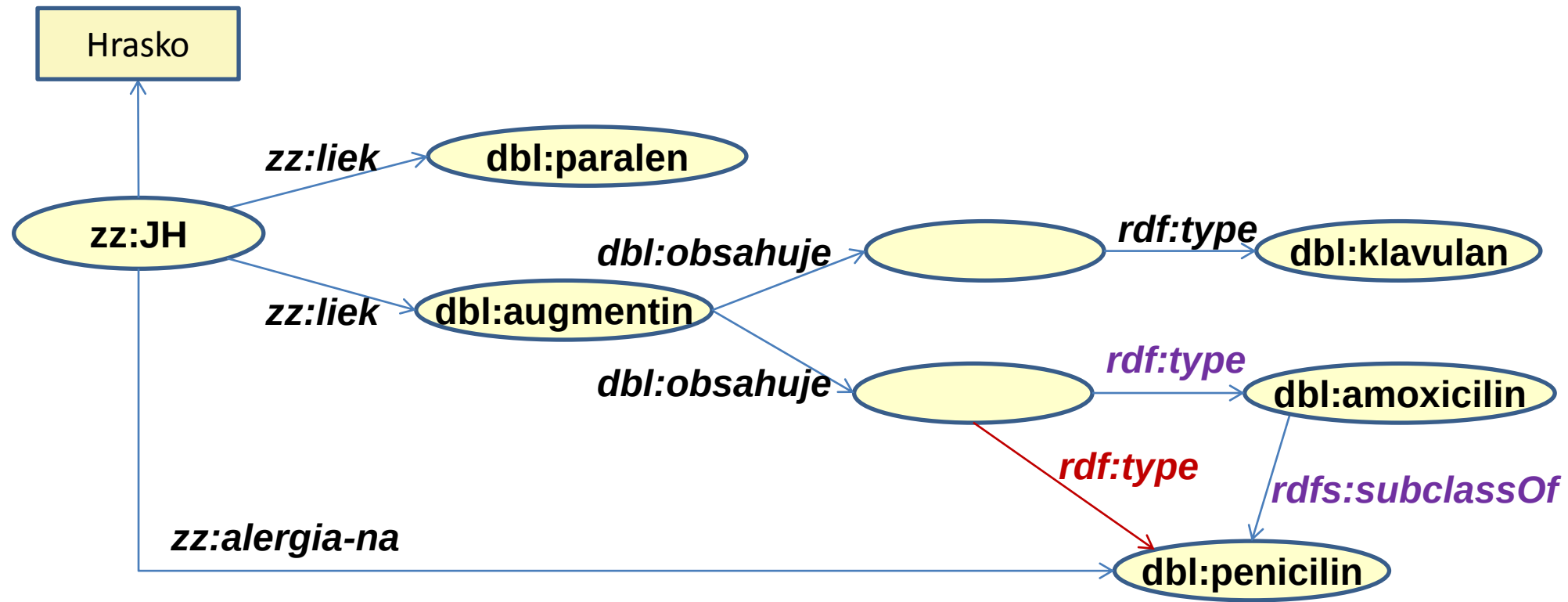


*s* rdf:type *x*  
*x* rdfs:subclassOf *y*  
-----  
*s* rdf:type *y*

# Dedukčné pravidlo pre rdf:type



# Dedukčné pravidlo pre rdf:type



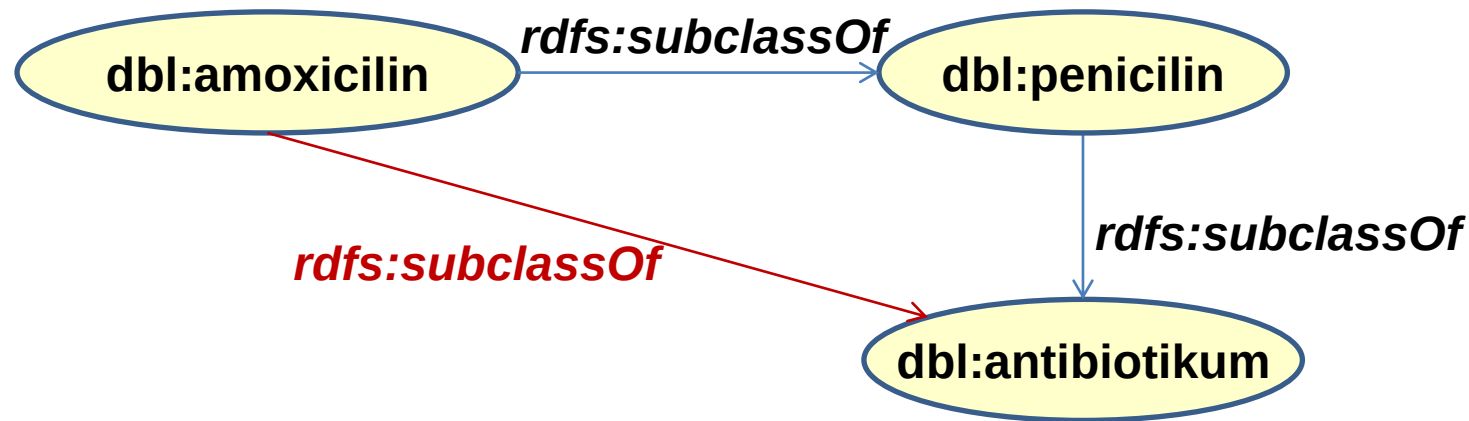
`s rdf:type x`  
`x rdfs:subclassOf y`

---

`s rdf:type y`



# Dedukčné pravidlo pre tranzitivitu `rdfs:subclassOf`



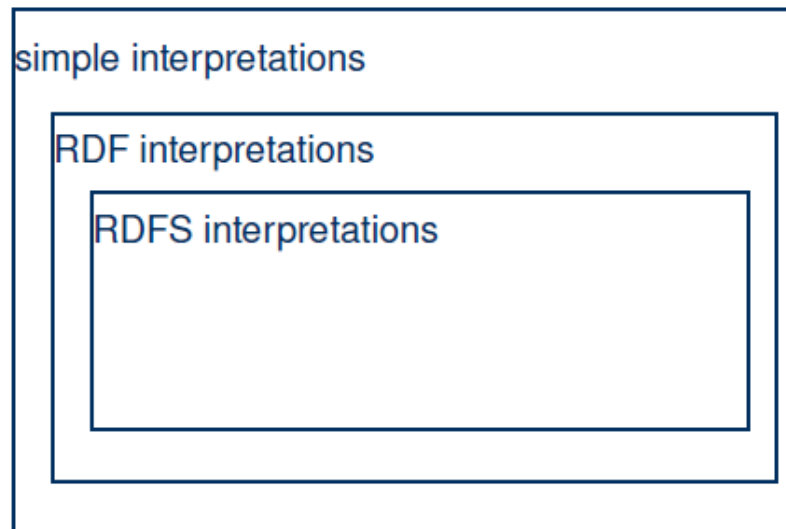
```
s rdfs:subclassOf x  
x rdfs:subclassOf y  
-----  
s rdfs:subclassOf y
```

# Triedy interpretácií

Ked sa zaoberáme interpretáciami nejakého formálneho jazyka, môžeme uvažovať o akýchkoľvek interpretáciách ale môžeme požadovať aby interpretácie mali isté vlastnosti. Termín **rdfs:subClassOf** môžeme obecné interpretovať ako akýkoľvek pojem napr. vzťah „**byť** **ziakom** **triedy**“. V slovníku RDFS však tento termín reprezentuje vzťah, ktorý sme si intuitívne interpretovali ako vzťah podmnožiny alebo podtriedy. Takáto interpretácia má dôležitú vlastnosť - **tranzitivitu**.

## Semantics of RDF(S)

- we proceed stepwise:



- the more we restrict the set of interpretations, the stronger the consequence relation becomes

# Triedy interpretácií

**Jednoduchá interpretácia** môže menám slovníka priradiť akýkoľvek význam:

- Univerzum môže byť akákoľvek množina prvkov, pojmov, vzťahov...
- Hodnoty zobrazenia  $I_S$  a  $I_L$  môžu byť akékoľvek objekty/pojem z univerza
- Hodnoty funkcie  $I_{EXT}$  akékoľvek binárne relácie nad univerzom

Nekladie na tieto množiny a zobrazenia žiadne podmienky či obmedzenia.

V ďalšom sa budeme zaoberať triedami interpretácií, ktoré spĺňajú určité **sémanticke podmienky – axiómy**, ktoré sa považujú za platné (v danej triede interpretácií). S využitím týchto axióm a sémantických podmienok je možné z daného predpokladu (RDF-grafu) dedukovať viac dôsledkov ako pri jednoduchej interpretácii.

Nech  $\mathcal{I}$  je nejaká trieda interpretácií, potom platí:

Ak  $E$  je jednoduchým dôsledkom  $G$ , potom je  $E$  dôsledkom aj v triede interpretácií  $\mathcal{I}$ .