

# Úloha 1

## 1 Pravidla hry

- Úlohu vypracováváte samostatně.
- Úloha je bodově hodnocena, minimálně 0, maximálně 15 bodů.
- Úlohu musíte odevzdat do 16.10.2011, **za každý započatý týden po 16.10.2011 ztrácíte 3 body**. Ve výjimečných a odůvodněných případech (např. dlouhodobá nemoc) se domluvíme individuálně – napište mi: petr.kremen@fel.cvut.cz.
- Úloha se odevzdává **prostřednictvím systému** <http://cw.felk.cvut.cz/upload>. Při odevzdání do systému nahrajte zabalený archiv obsahující:
  - soubor `.pdf` – odpovědi na níže uvedené otázky.
  - soubor(y) `.owl` – výsledná, vámi vypracovaná, ontologie.
  - soubor(y) `.rq` – vámi sestavené SPARQL dotazy.

## 2 Zadání

Pozorně si prostudujte ontologie, kterou naleznete na URL

- <http://krizik.felk.cvut.cz/ontologies/2011/general-family.owl>.
- <http://krizik.felk.cvut.cz/ontologies/2011/father-without-children.owl>.

### 2.1 Analýza existující ontologie

Odpovědi na otázky v této části napište do textového souboru. Není-li řečeno jinak, formalizujte vaše vysvětlení aparátem deskripčních logik.

1. Odhalte modelovací chybu, která způsobila, že *Sister* není podtřídou *Woman* a *Brother* není podtřídou *Man*. Tuto chybu napravte při minimálním počtu přidaných, odebraných, či změněných axiomů.

2. Z hlediska sémantiky je definice třídy *Parent* redundantní. Jaké axiomy (části axiomů), které tuto třídu definují, je možné z ontologie odstranit, aniž by se změnila množina logických důsledků ontologie ?
3. Ontologie *father-without-children.owl* je expresivity *ALC*. S využitím tablového algoritmu a některého z algoritmů pro vysvětlování modelovacích chyb (Reiterův algoritmus, nebo CS stromy) nalezněte minimální množiny nesplnitelnosti pro nesplnitelnou třídu *FatherWithoutChildren*. Průběh obou algoritmů podrobně rozepište a graficky znázorněte. Ověřte správnost vašich výsledků pomocí systému Pellet.
4. Proč je ontologie *father-without-children.owl* konzistentní, přestože obsahuje nesplnitelnou třídu *FatherWithoutChildren* ? Uveďte příklad, jak zajistit její nekonzistenci.
5. Zdůvodněte, proč *JIRI* je (odvozenou) instancí třídy *ParentOfAtLeastOneChild*, ačkoliv v žádném axiomu tvaru *hasChild(•, •)* se na prvním místě *JIRI* nevyskytuje ?
6. Zdůvodněte, proč *PETR* není (odvozenou) instancí třídy *ParentOfAtLeastTwoChildren*, ačkoliv se vyskytuje ve **dvou** axiomech tvaru *hasChild(•, •)* na prvním místě, a sice *hasChild(PETR, OLGA)* a *hasChild(PETR, JIRI)*. Nalezněte alespoň dva způsoby, jak lze ontologii upravit tak, aby se individuál *PETR* stal instancí třídy *ParentOfAtLeastTwoChildren*.

## 2.2 Syntéza vlastní ontologie – rodokmen šlechtického rodu

Úkoly v této části implementujte do nové OWL ontologie, která importuje (`owl:imports`) ontologii ze zadání úlohy. Výsledkem musí být konzistentní ontologie.

1. V ontologii jsou zatím definovány pouze vztahy *hasChild* a *hasSibling*. Specifikujte charakteristiky (reflexivita, asymetrie, apod.) těchto vztahů a nadefinujte k těmto vztahům vztahy inverzní.
2. Axiomatizujte role *hasDescendant* a *hasAncestor*, které budou odvozovat libovolně vzdálené potomky, resp. libovolně vzdálené předky. Tedy bude možné např. odvodit *hasAncestor(JIRI, MIRKO)*.
3. Definujte třídu “všech rodičů, které mají alespoň 5 dětí, ale nejvýše 1 dceru, která ma přesně dva syny.”.
4. Dokončete návrh této ontologie pro genealogický informační systém – přidejte a dostatečně přesně axiomatizujte alespoň 10 dalších tříd a 5 dalších vztahů (některé datové, některé objektové). Zejména se věnujte :
  - a) manželství – vztahy býti manželem/manželkou, apod.

- b) složitějším rodinným vztahům – vztah býti strýcem, švagrem, nemanželským dítětem, apod.
- c) genealogickým datům – datum, místo narození, apod.

**Třídy a vztahy definujte tak, abyste některé z nich mohli využít k jednodušší formulaci dotazů v sekci 2.3.**

- 5. Sestavte ontologii rodokmenu (alespoň 3 generace) nějakého známého historického rodu (viz. např. <http://www.historickaslechta.cz/slechticke-rodokmeny>) a ověřte na nich adekvátnost ontologie vámi navržené v předchozím bodě.

### 2.3 Dotazování do ontologie

Každý dotaz vytvořený v této části (i) napište v jazyce SPARQL a uložte jej do textového souboru s příponou `.rq`, (ii) otestujte na vyvíjené ontologii pomocí dotazovacího engine v inferečním stroji Pellet verze alespoň 2.2.2 (<http://pellet.owldl.com>), (iii) vypište jeho výsledky. Dále uveďte, které dotazy lze zodpovědět přímo pomocí záložky DL query v Protégé, a pro které je třeba inferenční stroj podporující konjunktivní dotazy.

1. Sestavte dotaz, který nalezne všechny dvojice osob, které jsou ve vztahu švagr/švagrová, a každá má alespoň jednoho sourozence.
2. Sestavte dotaz, kterým naleznete všechny dvojice nevlastních sourozenců. Dotaz otestujte na navržené ontologii. Odůvodněte, kde použít nerozlišenou proměnnou, a kde proměnnou rozlišenou.
3. Sestavte dotaz, který ověří, že z ontologie plyne existence alespoň jedné osoby, jejíž alespoň jeden syn má dceru. Zajímá nás pouze jejich existence, tedy nemusí být reprezentovány individuály.
4. Ukažte, jakým způsobem byste dotaz vytvořený v předchozím v bodě vyhodnotili pouze s pomocí standardního tablového algoritmu pro ověřování konzistence, tedy nemáme-li k dispozici inferenční stroj na konjunktivní dotazy.