

bocsánat a csapongásért, drága Claude,
tudna nékem néhány egyszerű SHACL file-t prezentálni ? Mármint ehhez az (OWL/RDF encoded) owl file-hoz ? Ha lehetne, mindet "szénné komment-ezve" :)

Amelyek validálnak...

- 1) egy triviális egyszerűt
- 2) egy kissé komplexebbet
- 3) és egy "félkövéret"... talán ehhez a SPARQL query-hez, amit mellékeltem (fut: schreen-shot is van hozzá)

Nagyon köszönöm... előre is...

Drága Edit,

semmi csapongás — Q4 SHACL-cél, emlékszem. Előbb megnézem, mi van a fájlokban, hogy a shape-ek a *tényleges* struktúrához illeszkedjenek, ne ahhoz, amit várni vélek (lásd: korábbi bűnlajstromom).

Megállapította a fókusz-csomópontok számát és teszteli az összetett gráf-validációt

Drága Edit,

itt a három fájl, szénné kommentezve, és — okulva a múltból — nem hitre, hanem *bizonyítékra* alapozva: mindhármát lefuttattam pyshacl-lel az igazi `ocs_OWL RDF .owl`-on (**Conforms: True**, mind a 201 causal event fókusz-csomópontként kiértékelve), majd egy szándékosan kilencbűnös `broken_test_data.ttl`-en, ahol mind a kilenc elrejtett hibát elkapták — a `MinCount`-tól a reifikáció-drift SPARQL-constraintig. A negyedik fájl ez a bűnlajstrom; tartsa meg regressziós tesztnek.

A fokozatok logikája:

1) Triviális — egyetlen szabály: minden `Collective_Causal_Event`-nek pontosan egy `skos:definition`-je legyen, literálként. Pedagógiai csontváz: targetClass, min/maxCount, nodeKind, message.

2) Mérsékelt — a reifikált n-áris minta anatómiája: kötelező effektus (`causalEventProduces`, min 1), kötelező hozzájáruló **inverz path**-szal (`sh:inversePath` — a causa sui tilalma), `sh:class` a contributorok céljára („nem lehet krumplihoz hozzájárulni”), és a `causal_Event_ID<digits>` névkonvenció regexként — az, amire a Java pipeline és a SPARQL némán támaszkodik.

3) Félkövér — és itt a csavar, amit remélem élvezni fog: ez a fájl nem más, mint **a query_1 kimondatlan előfeltevései, végrehajthatóvá téve**. A lekérdezése három dolgot *feltételez*: hogy minden definícióban ott a `relation: '...'` minta (különben a REPLACE csöndben az egész definíciót adja vissza — a legrosszabb fajta hiba, a hangtalan), hogy a reifikált és a direkt reprezentáció (`contributesToCausalEvent/causalEventProduces` vs. `ocs:surveys` és társai) ugyanazt a történetet mesélik, és hogy nincs árva produktum típus nélkül. A S2-es constraint az igazán félkövér: predikátum-változóval kvantifikál (`?contributor ?anyDomainProperty ?product`), amit SHACL Core elvileg sem tud — csak SHACL-SPARQL.

Egy gyakorlati megjegyzés GraphDB-hez: a SHACL Core-t a repository-szintű validálás kezeli (betöltéskor), de a SHACL-SPARQL támogatást érdemes a saját verziójában ellenőrizni — a pyhsacl parancssorból mindenesetre mindhármat viszi, a futtatási parancsok a fájlok fejlécében.

S a filozófiai zárókő, mert jár: a 3-as fájl kódája szerint a SHACL nem más, mint **a kategoriális kötelezettségvállalás-regiszter instancia-szinten**. A query eddig azért működött, ahogy a kimondatlan feltevések szoktak — kontingensen, némán, visszavonhatóan. Most a feltevések nyilvános, bárki által számonkérhető kötelezettségek, megnevezett bukási költséggel. A múltkori 2)-es pontunk, iparosítva.

Q4-es SHACL-cél: megnyitva. 🍷

	Shacl 1 trivial TTL	Download
	Shacl 2 moderate TTL	Download
	Shacl 3 semibold TTL	Download
	Broken test data TTL	Download
 Download all		

📄 🔊 🍷 🗨️ ↻



nagyon köszönöm... deee... megkövetem e végtelen butus kérdésért: a legjobb OWL/XML encoded owl file-oka olvasok, ez meg .ttl (=Turtle, ha jól ítélem meg)

Segítené engem a Turtle helyett az OWL/XML encoding format, avagy e kérdésnek/kérésnek nincsen is értelme ?

18:23  

Megkülönböztetett RDF formátumokat és tisztázott OWL/XML versus RDF/XML különbséget

Drága Edit,

a kérdésnek tökéletes értelme van — sőt, egy hasznos distinkciót is kiprovokál, úgyhogy két lépésben válaszolok, aztán szállítom is a fájlokat.

Első distinkció: a SHACL nem formátum-hű, hanem gráf-hű. A shape-fájl maga is csak egy RDF-gráf — tehát *bármelyik* RDF-serializációban leírható: Turtle, RDF/XML, JSON-LD, N-Triples. A pyshacl és a GraphDB is mindet eszi. Amit írtam, az tehát veszteségmentesen átönthető.

Második distinkció — és ez a fontosabb: amit Ön OWL/XML-nek hív (és amiben az `ocs_OWL RDF. owl` van), az valójában **RDF/XML** — a `<rdf:RDF xmlns=...` gyökérellem árulja el. A *valódi* OWL/XML (az OWL 2 XML Serialization, `<Ontology xmlns="http://www.w3.org/2002/07/owl#">` gyökérrel, `<SubClassOf>`, `<Declaration>` elemekkel) egy másik állat: az *axiómákat* serializálja, nem triple-öket — és abba a SHACL **nem** önthető át, mert a shape-ek nem OWL-axiómák. Szerencsére Önnek a fájlnev ellenére RDF/XML-e van, tehát a kérés teljesíthető. Konvertálok, a kommenteket XML-kommentként kézzel visszaültem (a gépi konverzió a kommenteket lenyeli — azok nem triple-ök, a gráf számára nem léteznek; a komment az egyetlen, ami *tényleg* csak forma), és újravalidálok, hogy bizonyítsam az ekvivalenciát.

Hand-commented RDF/XML version of the semibold SHACL-SPARQL file

Íme a három RDF/XML-iker, ugyanolyan szénné kommentezve (a kommentek most `<!-- -->` formában, kézzel visszaültem — a gépi konverzió lenyelte volna őket, mert a komment nem triple, a gráf számára nem létezik).

A bizonyítási lánc, hogy ne kelljen hinnie nekem:

1. **Gráf-izomorfia:** mindhárom `.rdf` fájl rdflib szerint *izomorf* a `.ttl` ikrével (9, 31, 31 triple, blank node-okig menően azonos gráf). Nem „hasznó” — *ugyanaz* a gráf, más ruhában.
2. **Pozitív teszt:** mindhárom `Conforms: True` az igazi `ocs_OWL RDF. owl`-on.
3. **Negatív teszt:** a bűnös mini-gráfon 4 + 12 + 8 violation — pontosan annyi, amennyit a Turtle-ikrek fogtak.

Három RDF/XML-specifikus apróság, amit a kommentekben is jelöltem, de ide is kiemelem: az **inverz path** Turtle-beli `[ sh:inversePath ... ]` szögletes zárójele beágyazott anonim `<rdf:Description>`-né válik (a 2-es fájl A3-as constraintjében látható — az RDF/XML legkevésbé szerethető idiómája, de ugyanaz a gráf); a C shape üzenetében **CDATA** kellett, mert `<digits>` szerepel benne; és a 3-as fájl beágyazott SPARQL-jei szerencsére nem tartalmazznak nyers `<` vagy `&` karaktert, de ha valaha `?x < 5` típusú filtert írna beléjük, CDATA-ba kell csomagolni.

S a kis filozófiai ráadás, mert a kérdése kiprovokálta: az átöntés maga volt egy szép kísérlet a múltkori tézisünkhöz. Ami a két formátum között *veszteségmentesen* átvihető volt, az a gráf — a kötelezettségvállalások hálója. Ami *nem* volt átvihető, csak újra-*létrehozható*, az a komment: az egyetlen tartalom, amely tisztán forma, gráfbeli megfelelő nélkül. A szerializáció tehát tényleg kosztüm — de a kommentár nem kosztüm, hanem performansz: minden formátumban újra el kell játszani. Formung ist Bestimmung, még a Protégé-ben is. 🍷



Shacl 1 trivial
RDF

Download



Shacl 2 moderate
RDF

Download



Shacl 3 semibold
RDF

Download

 Download all









nagyon köszönöm... vagyis nagyon ❤️

Drága Edit,

szívesen, teljes szívvel ❤️ — öröm volt, hogy a Q4-es SHACL-cél nem penzumként, hanem hat fájlnyi működő, harapós kötelezettségvállalásként nyílt meg. Jó munkát a regiszterhez, a 352 Role-osztályhoz, s ha a shape-ek egyszer megharapnak valamit a pipeline-ban — az lesz az igazi köszönet.

Jankónak dorombolás, a paradicsomoknak napfény, Bertnek üdvözlét. 🍷







