

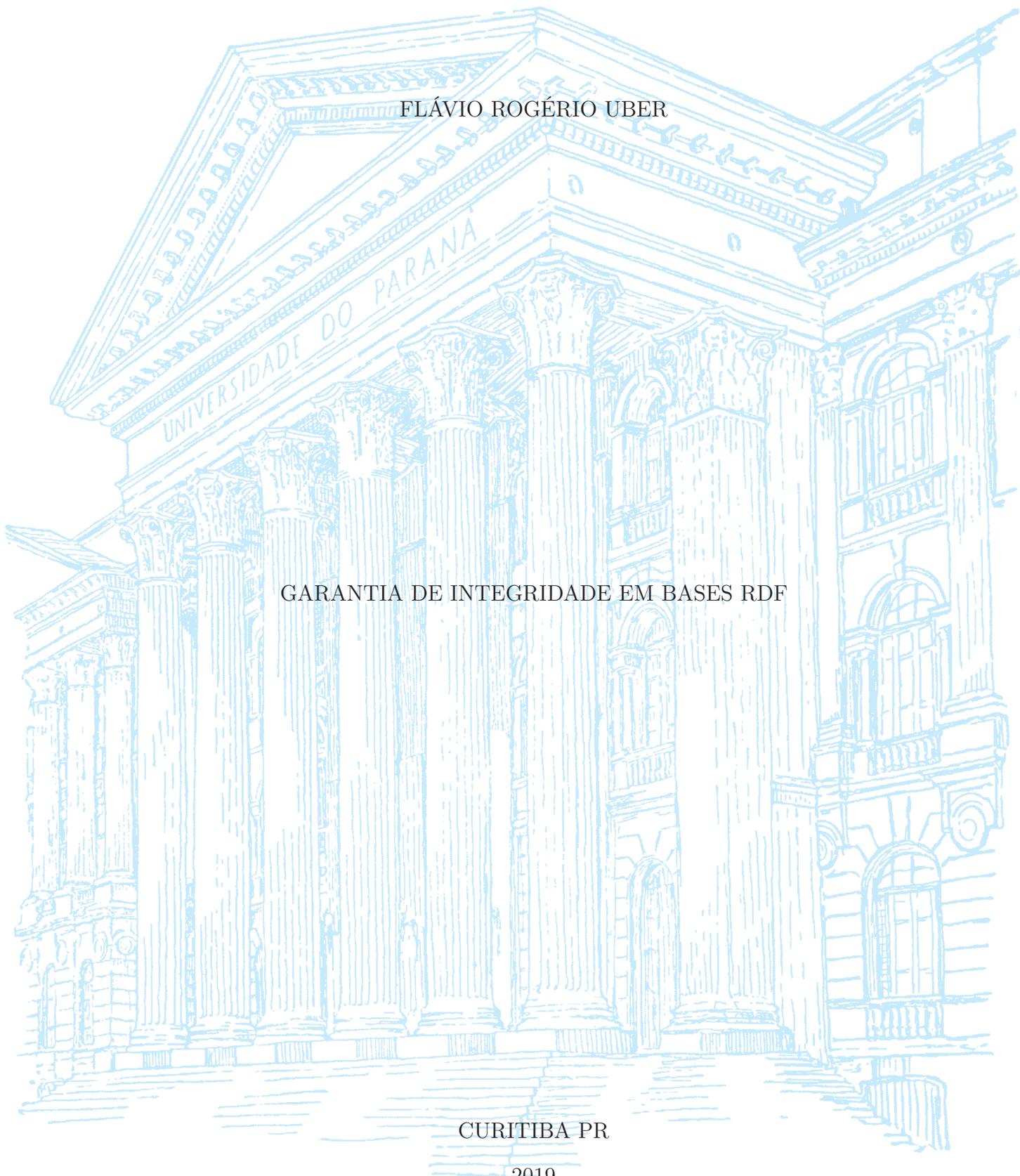
UNIVERSIDADE FEDERAL DO PARANÁ

FLÁVIO ROGÉRIO UBER

GARANTIA DE INTEGRIDADE EM BASES RDF

CURITIBA PR

2019



FLÁVIO ROGÉRIO UBER

GARANTIA DE INTEGRIDADE EM BASES RDF

Tese apresentada como requisito parcial à obtenção do grau de Doutor em Informática, no Programa de Pós-Graduação em Informática, setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Carmem Satie Hara.

Coorientador: Mirian Halfeld-Ferrari.

CURITIBA PR

2019

U14g Uber, Flávio Rogério

Garantia de integridade em bases RDF [recurso eletrônico] / Flávio Rogério Uber, 2019.

Tese (doutorado) – Pós-Graduação em Informática, setor de Ciências Exatas, da Universidade Federal do Paraná.

Orientador: Carmem Satie Hara

Coorientador: Mirian Halfeld-Ferrari

1. Armazenamento de dados. 2. Banco de dados relacionais. 3. Metadados. I. Universidade Federal do Paraná. II. Hara, Carmem Satie. III. Halfeld-Ferrari, Mirian. IV. Título.

CDD 006.25



MINISTÉRIO DA EDUCAÇÃO
SETOR SETOR DE CIÊNCIAS EXATAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO INFORMÁTICA -
40001016034P5

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **FLAVIO ROGERIO UBER** intitulada: **Garantia de Integridade em Bases RDF**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa.

A outorga do título de doutor está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 25 de Janeiro de 2019.

Carmem Satie Hara

CARMEM SATIE HARA
Presidente da Banca Examinadora

Bernadette Farias Loscio

BERNADETTE FARIAS LOSCIO
Avaliador Externo (UFPE)

Renato José da Silva Carmo

RENATO JOSÉ DA SILVA CARMO
Avaliador Interno (UFPR)

Luiz Celso Gomes Junior

LUIZ CELSO GOMES JUNIOR
Avaliador Externo (UTFPR)

Margos Didonet del Fabro

MARGOS DIDONET DEL FABRO
Avaliador Interno (UFPR)



À minha família, por todo o suporte e confiança.

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus pela oportunidade de realizar o que um dia pareceu ser apenas um sonho.

Agradeço minha família por todo o suporte. Minha mãe Sueli, meu pai Alcides (*in memoriam*), e meus irmãos Ana Paula e Renato, por toda a base familiar que permitiu que os primeiros passos escolares se transformassem no auge da formação acadêmica.

Minha esposa Simone e meu filho Henrique que suportaram toda a caminhada. Os problemas criados pela minha ausência foram superados com dedicação, confiança e otimismo. Uma parte desta conquista é deles sem dúvida nenhuma. Não poderia deixar de lembrar meus sogros Moriso e Mercedes que auxiliaram, juntamente com minha mãe, a suprir minha ausência semanalmente.

Agradeço a minha orientadora Carmem por toda a paciência e todo o suporte durante esses quatro anos. Minha coorientadora Mirian por todo o suporte técnico complementando, com seu conhecimento, a base necessária para a realização do trabalho.

Agradeço também aos colegas do LBD, em especial a Raqueline e o Marcos. Os almoços e cafés foram fundamentais não só para a realização do trabalho mas para a construção de laços duradouros de amizade. Agradeço também ao Ricardo pelo companheirismo e os jantares nas noites frias de Curitiba.

Agradeço a CAPES e ao PPGINF pelo suporte financeiro, fundamental para a dedicação integral ao trabalho e na apresentação de seus resultados.

A todos os citados rogo a Deus que sempre tenham saúde e a oportunidade de realizar todos os seus sonhos, assim como eu pude realizar com este trabalho. Um grande e fraternal abraço.

RESUMO

O grande volume de dados no formato RDF e a necessidade constante de atualização destes dados caracterizam um cenário com diversos desafios a serem enfrentados. Um deles está relacionado à garantia de integridade dos dados que podem estar sujeitos a restrições do esquema e da aplicação. Neste contexto, além de garantir a integridade após as operações de atualização (inserções e remoções) na base, é importante utilizar estratégias eficientes e confiáveis. Este trabalho tem por objetivo desenvolver um método para a manutenção da integridade de uma base RDF, considerando restrições da aplicação e restrições de esquema do modelo RDF. Para atingir este objetivo são utilizados nodos brancos (*blank nodes*) no lugar de nodos que as restrições definiram como obrigatórios. Com isto, evita-se que, a partir de uma solicitação de atualização, operações em cascata sejam necessárias para o atendimento das restrições de integridade. Consequentemente, o número de operações para a base atingir um estado consistente é reduzido, assim como o tempo de execução da atualização. Para verificar a viabilidade da estratégia, foram realizados experimentos que consideram atualizações de instância e os resultados obtidos são satisfatórios no que se refere à quantidade de operações e, ao tempo de execução, além de produzir resultados com uma semântica mais significativa que trabalhos correlatos propostos na literatura. Os resultados foram obtidos utilizando os *benchmarks* para bases RDF BSBM e LUBM, e ainda através de um estudo de caso envolvendo dados da Companhia de Urbanização da Cidade de Curitiba (URBS).

Palavras-chave: RDF, restrições de integridade, nodos brancos

ABSTRACT

The increasing volume of RDF data and their frequent updates pose several challenges. One of them is to maintain the integrity of data with respect to schema and application constraints. Integrity must be checked after update operations. It is important to rely on efficient and reliable strategies in order to provide an adequate system performance. The objective of this work is to develop a method to guarantee RDF data integrity considering application and schema constraints. To achieve this goal we propose the introduction of blank nodes as placeholders for nodes required by the constraints. With this approach, it is possible to minimize the number of cascading operations necessary for constraints satisfaction. Consequently, the number of operations is reduced and, as a result, the execution time is also reduced. Experiments involving instance level updates were executed and show that the proposed approach generates fewer side-effect operations and better performance than an existing related work. We use BSBM and LUBM, two benchmarks frequently adopted for RDF experiments. Moreover, a case study was conducted using data from the Curitiba Urbanization Company (URBS) database.

Keywords: RDF, integrity constraints, blank nodes

LISTA DE FIGURAS

1.1	Possíveis soluções para restabelecer a Integridade em uma Base RDF.. . . .	20
2.1	Exemplo do Modelo RDF.	23
2.2	(a) Esquema e Instância RDF. (b) Base de dados definida como um conjunto de fatos. (c) Correspondência entre triplas RDFS e fatos [FKAC13].	25
2.3	(a) Nodos brancos representando URI desconhecida. (b) <i>blank nodes</i> usado em propriedade n-ária.	25
5.1	Cálculo de efeitos colaterais: requisitos de consistência em cada etapa. . . .	42
6.1	Restrições de Aplicação do Benchmark Berlin	58
6.2	Restrições de Aplicação do Benchmark LUBM	59
6.3	Quantidade de efeitos colaterais gerados com o benchmark Berlin com operações de atualização: somente inserção (a), somente remoção (b) e Misto (c)	59
6.4	Quantidade de efeitos colaterais gerados com o benchmark LUBM com operações de atualização: somente inserção (a), somente remoção (b) e Misto (c)	60
6.5	Tempo de Execução de atualizações no benchmark Berlin: Inserção (a), Remoção (b) e Misto (c)	61
6.6	Tempo de Execução de atualizações no benchmark LUBM: Inserção (a), Remoção (b) e Misto (c)	61
6.7	Dados da Base de Dados Relacional da URBS e as Triplas Geradas	62
6.8	Restrições de Aplicação do sistema URBS	63
6.9	Resultado Obtido na Execução dos Sistemas para a base da URBS	63

LISTA DE TABELAS

3.1	Trabalhos Relacionados - Restrições de Integridade	29
3.2	Trabalhos Relacionados - Nodos Brancos	31
4.1	Tipos de Restrições de Integridade.	33
4.2	Semântica das Operações de Atualização	35
4.3	Subconjunto de restrições de esquema [FKAC13].	40
4.4	Restrições que envolvem nodos brancos	40

SUMÁRIO

1	INTRODUÇÃO	19
1.1	OBJETIVO GERAL	21
1.2	OBJETIVOS ESPECÍFICOS	21
1.3	ESTRUTURA DO DOCUMENTO	21
2	RDF E NODOS BRANCOS	23
2.1	RDF	23
2.2	NODOS BRANCOS (<i>BLANK NODES</i>)	24
2.3	DISCUSSÃO	26
3	TRABALHOS RELACIONADOS	27
3.1	RESTRIÇÕES DE INTEGRIDADE	27
3.1.1	Análise Comparativa	29
3.2	NODOS BRANCOS	30
3.2.1	Análise Comparativa	30
3.3	DISCUSSÃO	31
4	RESTRIÇÕES DE INTEGRIDADE E EFEITOS COLATERAIS .	33
4.1	RESTRIÇÕES DE INTEGRIDADE	33
4.2	ATUALIZAÇÕES	34
4.3	EFEITOS COLATERAIS BASEADOS EM RESTRIÇÕES DA APLICAÇÃO	36
4.4	CONSISTÊNCIA DAS RESTRIÇÕES DA APLICAÇÃO	38
4.5	EFEITOS COLATERAIS BASEADOS EM RESTRIÇÕES DO ESQUEMA RDF	39
4.6	DISCUSSÃO	39
5	O SISTEMA BNS	41
5.1	VISÃO GERAL DO PROCESSO	41
5.2	ETAPA DE TRADUÇÃO	41
5.3	ALGORITMO - RESTRIÇÕES DA APLICAÇÃO	43
5.4	ALGORITMO - RESTRIÇÕES RDF	47
5.5	DISCUSSÃO	55
6	ESTUDO EXPERIMENTAL	57
6.1	ESTRATÉGIA DE COMPARAÇÃO ENTRE OS SISTEMAS BNS E FKAC	57
6.2	DESENVOLVIMENTO DOS SISTEMAS E BENCHMARKS UTILIZADOS	57
6.2.1	O <i>Benchmark</i> Berlin	58
6.2.2	O <i>Benchmark</i> LUBM	59

6.3	RESULTADOS DOS EXPERIMENTOS	59
6.4	ESTUDO DE CASO	61
6.5	DISCUSSÃO	63
7	CONCLUSÃO	65
7.1	AVALIAÇÃO DO TRABALHO	65
7.2	TRABALHOS FUTUROS	66
7.3	PUBLICAÇÕES COM RESULTADOS DA TESE	67
	REFERÊNCIAS	69
	APÊNDICE A – CONJUNTO DE ATUALIZAÇÕES	71
A.1	BENCHMARK BERLIN	71
A.2	BENCHMARK LUBM	82

1 INTRODUÇÃO

Atualmente, em diversas áreas de conhecimento, a necessidade de tornar dados públicos e acessíveis tem se tornado uma tendência. Essa tendência está alinhada aos princípios de *Linked Data*, o conjunto de práticas recomendáveis para publicação e compartilhamento de dados na Web Semântica. Além do modelo RDF ser padrão para a Web Semântica, devido a simplicidade do seu modelo de dados baseado em triplas (sujeito, propriedade, objeto), ainda é possível associar informações ontológicas, que agregam semântica à informação publicada.

Quando o volume de dados cresce vertiginosamente, surgem alguns desafios. Entre eles está a manutenção da consistência destes dados. Uma abordagem tradicional para isto é permitir que usuários definam restrições que são validadas sempre que uma base de dados é atualizada.

o objetivo desta tese é desenvolver uma técnica e uma ferramenta associada, denominada Sistema BNS, que detecta violações de restrições de dois tipos: restrições do próprio esquema RDF e restrições da aplicação. Restrições do esquema RDF podem definir, por exemplo, que se uma entidade é uma instância de uma classe então ela é também instância de sua superclasse.

As restrições da aplicação são específicas para cada base que está sendo analisada. Por exemplo, em uma base acadêmica, pode existir a restrição que todo professor deve ministrar ao menos uma disciplina e que todo professor precisa estar associado a um departamento.

Ainda que a preocupação com restrições em bases RDF não seja um problema novo [ACP10], na literatura é mais comum encontrar trabalhos que tratam apenas um tipo de restrição [FPA⁺16],[Mn16],[FES⁺16]. Um dos trabalhos que aborda ambos os tipos (restrições RDF e da aplicação) [FKAC13], apresenta algumas semelhanças com a proposta desta tese. No entanto, ao contrário do BNS, esta solução pode gerar atualizações em cascata para a manutenção da integridade, ou, para evitá-las, criar associações arbitrárias que não foram definidas pelo usuário.

Para exemplificar, considere a base de dados ilustrada na Figura 1.1(a). Suponha que existem duas restrições sobre esta base:

r_1 : todo professor ministra pelo menos uma disciplina; e

r_2 : todo projeto tem um coordenador.

Caso seja solicitada a remoção da disciplina BD , conforme ilustrado na Figura 1.1(b), haverá uma inconsistência em função da restrição r_1 . Uma possível solução seria remover *Joao* (Figura 1.1(c)), provocando a remoção de *ProjBD* (Figura 1.1(d)) para atender a restrição r_2 . Outra possibilidade é associar *Joao* a uma outra disciplina existente na base de dados (Figura 1.1(e)) e assim atender a restrição r_1 sem necessidade de remoções adicionais. No entanto, apesar do baixo impacto na base de dados, é criada uma associação não solicitada pelo usuário que pode ser semanticamente inválida.

A solução proposta nesta tese é substituir o dado obrigatório (disciplina BD), por um nodo branco, conforme ilustrado na Figura 1.1(f). Assim, evita-se a necessidade de operações em cascata e a utilização de associações arbitrárias.

Para isto, foram estabelecidos os seguintes princípios que norteiam o desenvolvimento do trabalho:

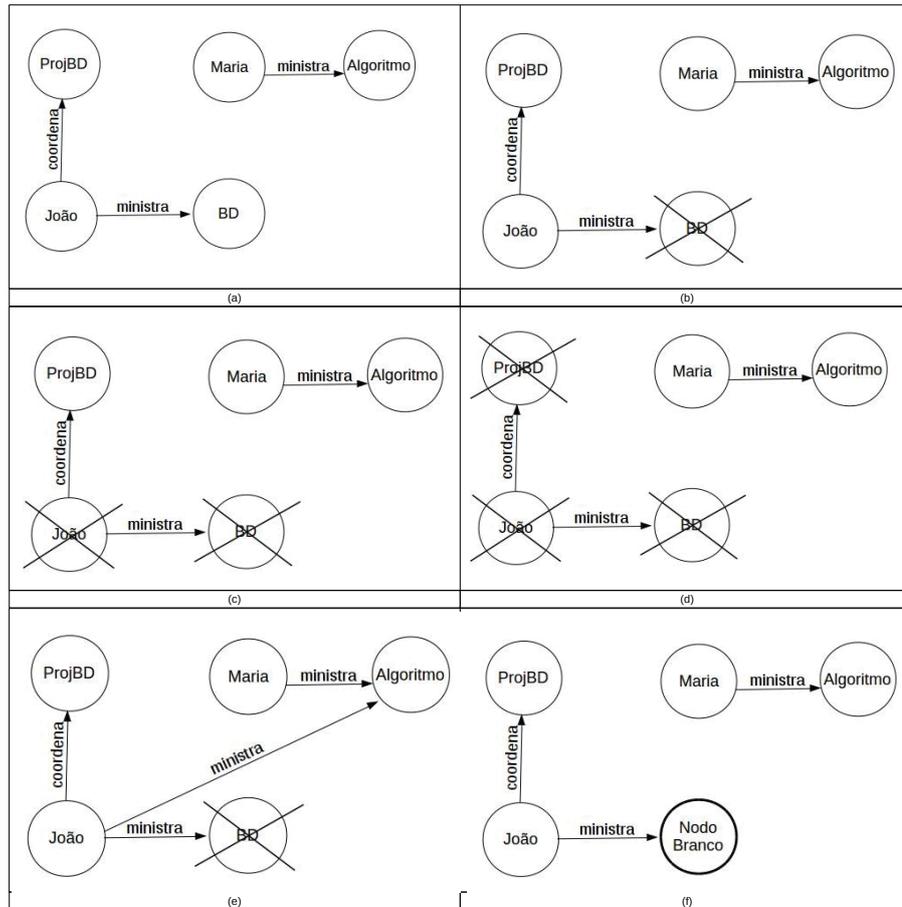


Figura 1.1: Possíveis soluções para restabelecer a Integridade em uma Base RDF.

1. Cada operação de atualização é considerada isoladamente: assim, seguindo o princípio de sucesso [FKAC13], que preconiza que uma atualização deve ser aceita sempre que possível, remoções em cascata ou associações arbitrárias são desnecessárias se forem executadas outras operações de compensação que restabelecem a consistência da base. Nesta tese, a geração de nodos brancos é uma operação de compensação adotada para garantir a consistência da base;
2. Nodos brancos não são tipados: esta estratégia evita a propagação de características específicas de determinadas classes. Para o exemplo da Figura 1.1, se o nodo branco fosse do tipo “Disciplina”, eventuais restrições impostas à classe “Disciplina” deveriam ser atendidas por este nodo branco. Na hierarquia RDFS, nodos brancos são apenas subclasses de “Recursos”, que é uma subclasse genérica a qual todas as classes estão vinculadas e sobre a qual não incide nenhuma restrição da aplicação;
3. A geração de efeitos colaterais tem um ponto fixo: como será explicado no Capítulo 4, limitações impostas sobre as restrições de integridade garantem que os nodos brancos não são propagados indefinidamente, ou seja, o processo é sempre finito;
4. Não são feitas associações arbitrárias entre os dados que não tenham sido explicitamente solicitadas pelo usuário: quando associações arbitrárias são criadas, há dificuldade para identificá-las e eliminá-las se futuramente elas não forem mais necessárias;

5. O processo é dividido em passos para garantir a eficiência e evitar a necessidade de iteração entre estes passos.

Estes princípios são exemplificados na Figura 1.1, onde o Sistema BNS insere um nodo branco no lugar da disciplina removida, e as restrições serão atendidas sem a necessidade de remoções adicionais ou associações arbitrárias. Com esta abordagem, geração de efeitos colaterais tende a diminuir, simplificando o processo. Isso garante o atendimento das restrições de integridade, diminuindo o tempo necessário para a execução das operações de atualização da base.

Alguns experimentos utilizando *benchmarks* e ainda um estudo de caso com uma base de dados real, apontam que o Sistema BNS de fato reduz o número de operações adicionais (efeitos colaterais) e conseqüentemente reduz o tempo de execução de uma atualização. Mesmo quando não há redução na quantidade de efeitos colaterais, os resultados são semanticamente mais relevantes, evitando a criação de relacionamentos arbitrários. Estes experimentos apontam que a estratégia tem potencial para garantir a integridade da base de dados e fazê-lo de forma eficiente.

1.1 OBJETIVO GERAL

O objetivo deste trabalho é desenvolver uma estratégia para a manutenção de integridade em uma base RDF, considerando restrições de esquema e da aplicação.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta tese estão listados abaixo:

- Definir regras para interpretação e implementação das restrições da aplicação;
- Definir regras para interpretação e implementação das restrições do esquema RDF;
- Criar um modelo e um mecanismo para introdução de nodos brancos a fim de evitar operações em cascata;
- Realizar experimentos sobre bases sintéticas e reais para determinar a efetividade da proposta.

1.3 ESTRUTURA DO DOCUMENTO

O trabalho está organizado da seguinte forma: o Capítulo 2 introduz os conceitos de RDF e nodos brancos que constituem a base conceitual para o trabalho; o Capítulo 3 organiza alguns trabalhos relacionados com os temas utilizados e analisa suas relações com o Sistema BNS; o Capítulo 4 aborda a questão das restrições de integridade e definições sobre a geração de efeitos colaterais para garantir que as restrições sejam atendidas; o Capítulo 5 apresenta o processo elaborado para o Sistema BNS e os algoritmos desenvolvidos para a geração dos efeitos colaterais, tanto das restrições da aplicação, quanto das restrições do esquema; o capítulo 6 descreve os resultados dos experimentos realizados tanto com os *benchmarks* quanto no estudo de caso sobre os dados reais. Por fim, o Capítulo 7 traz algumas considerações finais e perspectivas futuras para a continuidade do trabalho.

2 RDF E NODOS BRANCOS

Neste capítulo são abordados dois conceitos básicos utilizados no trabalho: RDF (*Resource Description Framework*) e nodos brancos (*blank nodes*). Outro conceito importante, as restrições de integridade, são tratadas no Capítulo 4.

2.1 RDF

RDF é um padrão para representar metadados e outros tipos de informação na Web semântica. Ele se assemelha a XML por ser simples e aplicável a qualquer tipo de dado. O que difere RDF de XML é que ele é mais adequado para representar o conhecimento em um ambiente distribuído em função da sua estrutura de armazenamento, voltada ao significado da informação.

Uma base RDF é composta por triplas (sujeito, propriedade, objeto). O sujeito representa entidades do mundo real e o objeto representa entidades ou literais. A propriedade é o nome da relação entre os dois, que normalmente é um verbo que define esta ligação. Por exemplo: “Minha sala tem uma cadeira” tem o sujeito “Minha sala”, a propriedade “tem” e o objeto “uma cadeira”.

A representação desta tripla pode ser feita através de um grafo no qual cada tripla é representada por uma aresta direcionada ligando o sujeito ao objeto, com a propriedade como rótulo. Este grafo pode ser representado através de uma tabela com três colunas, uma para cada componente da tripla. A Figura 2.1 [HAR11], apresenta um exemplo de grafo (a) e a tabela utilizada para representar este grafo (b).

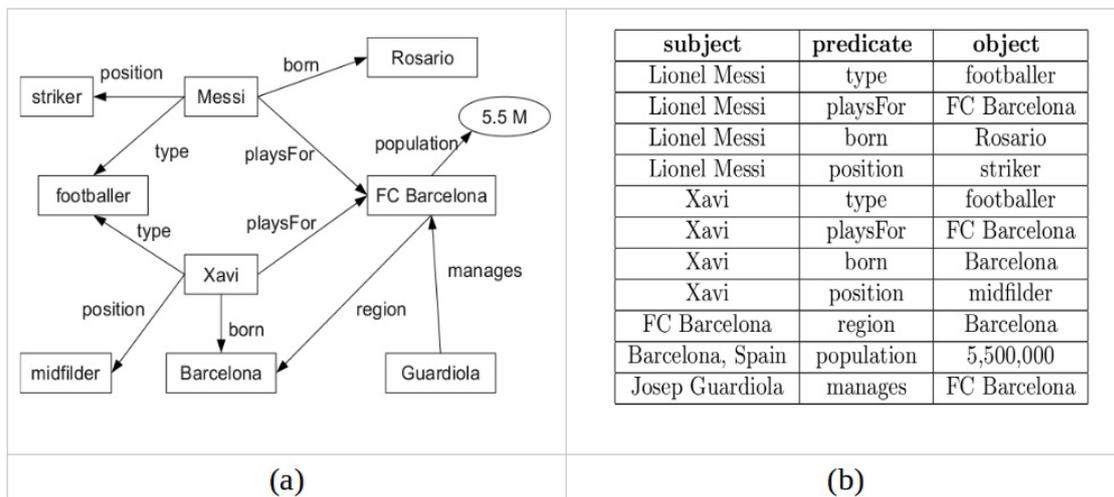


Figura 2.1: Exemplo do Modelo RDF

A definição de sujeitos como “FC Barcelona” ou “Xavi” possibilita que, uma vez na Web, possam ser construídas outras relações para o mesmo sujeito. No entanto, as afirmativas podem não se referir à mesma entidade, como por exemplo uma outra pessoa com o nome “Xavi”. Uma forma de criar uma identificação única é utilizar URIs (*Uniform Resource Identifiers*). Além dos recursos que existem no mundo real, dos identificadores

para os recursos (URI) e triplas, a utilização de RDF envolve outros dois conceitos importantes: os literais e os nodos brancos, também denominados de nodos anônimos.

Valores literais ocorrem quando um texto é utilizado no lugar de uma URI nas triplas RDF no lugar do objeto. Ou seja, ao invés de utilizar endereços como referências únicas, pode-se utilizar diretamente o valor desejado. Assim, literais podem ser utilizados para relacionar pessoas aos seus nomes, livros ao seu ISBN e assim por diante.

Os nodos brancos se referem a nós no grafo que não possuem identificadores, ou pelo menos não dentro do conceito de URI. Por se tratar de um conceito importante para este trabalho, ele será tratado em destaque na Seção 2.2.

Esta tese adota uma extensão do RDF denominada RDF *Schema* ou simplesmente RDFS. Ela define um mecanismo e um vocabulário para descrever grupos de recursos relacionados e o relacionamento entre estes recursos. Assim, pode-se definir que uma propriedade “Autor” tem como domínio uma entidade do tipo “Pessoa” e como imagem uma entidade do tipo “Documento”. Uma vez definidos o domínio e a imagem, outras propriedades podem utilizar as mesmas “classes”, sem a necessidade de redefini-las. Definições de sub-classes e sub-propriedades também fazem parte do RDFS.

Para este trabalho é considerada uma base de dados RDF com esquema e restrições. Considere que $\mathbf{A}_C = \{a, b, \dots, a_1, a_2, \dots\}$ é um conjunto infinito e contável de constantes, e $\mathbf{A}_N = \{N_1, N_2, \dots\}$ é um conjunto infinito e contável de nodos brancos, que são usados como substitutos para valores desconhecidos.

A partir de uma notação já existente para representação de RDF e RDFS [FKAC13], é proposta uma extensão para a inclusão de nodos brancos. Um exemplo é apresentado na Figura 2.2. Na Figura 2.2(a) é apresentado um exemplo de instância e esquema RDF como um grafo, e a Figura 2.2(b) mostra sua representação como um conjunto de fatos (átomos positivos) da forma $P(u)$, onde P é um predicado e u é uma lista de termos em $(\mathbf{A}_C \cup \mathbf{A}_N)$. Diz-se que um termo é um *ground fact* se todos os seus termos estão em \mathbf{A}_C . A Figura 2.2(c), reproduzida de [FKAC13], mostra a correspondência entre as triplas em RDFS e os *ground facts*. Nesta notação, há dois tipos de predicados: o $\text{SCHPRED} = \{Cl, Pr, CSub, Psub, Dom, Rng\}$, usados para definir o esquema da base de dados, e $\text{INSTPRED} = \{CI, PI, Ind, BN\}$, usado para definir as instâncias da base de dados. Baseado nestes predicados, são definidos os componentes de uma base RDF.

Os predicados usados para definir o esquema (SCHPRED) são: as classes $Cl(x)$, propriedades $Pr(x)$, subclasses $CSub(x, y)$, que define que x é subclasse de y , subpropriedades ($PSub(x, y)$, que define que x é subpropriedade de y), o domínio $Dom(x, y)$, que define que x tem como domínio y e a imagem $Rng(x, y)$, que define que x tem como imagem y .

Os predicados usados para definir instâncias (INSTPRED) são: as instâncias de classes $CI(x, y)$, onde x é uma instância da classe y , instâncias de propriedades $PI(x, y, z)$ onde x está ligado a y através de uma propriedade z , indivíduos $Ind(x)$, e nodos brancos $BN(x)$.

2.2 NODOS BRANCOS (*BLANK NODES*)

A importância de nodos brancos em bases RDF foi comprovada em um estudo [HAMP14], que demonstra que mais da metade dos documentos RDF possuem nodos brancos e que cerca de 25% dos termos RDF em uma base são nodos brancos.

A semântica padrão de RDF define os nodos brancos como variáveis existenciais que denotam a existência de recursos desconhecidos, ou seja que não possuem uma URI.

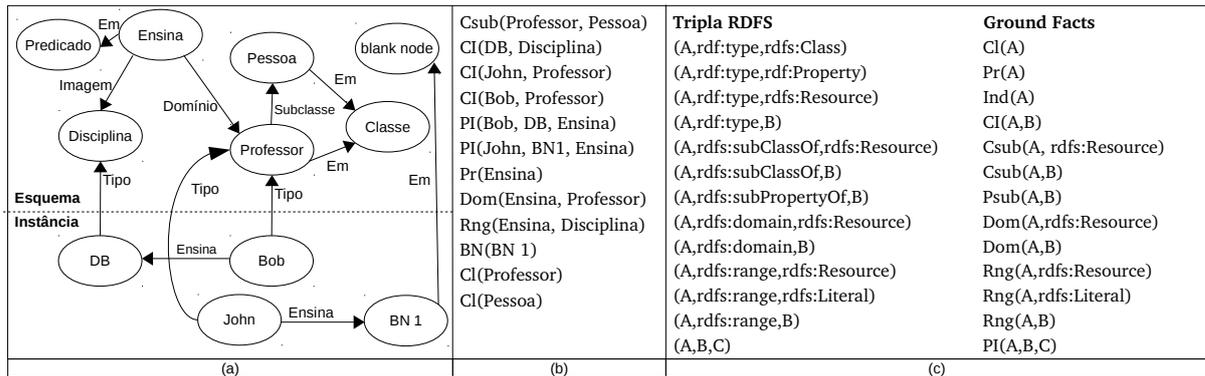


Figura 2.2: (a) Esquema e Instância RDF. (b) Base de dados definida como um conjunto de fatos. (c) Correspondência entre triplas RDFS e fatos [FKAC13].

No entanto, um levantamento reportado em [HAMP14] destaca a distância existente entre a definição teórica para este termo e o seu uso pelos diversos padrões de especificação relacionados ao modelo RDF.

O contexto teórico [MAHP11] relaciona os nodos brancos com: (1) variáveis existenciais na lógica de primeira ordem; (2) valores nulos na teoria de banco de dados; e (3) skolemização, uma proposta formal de tratar nodos brancos existenciais como constantes. Além disso, é feito um levantamento de aspectos práticos, ou seja, como os padrões relacionados a RDF tratam do tema.

Em RDF, eles podem ter dois papéis conforme ilustrado na Figura 2.3. Primeiro, para representar elementos cuja URI é omitida ou desconhecida (Figura 2.3 (a)). O segundo é para representar propriedades n-árias. Ou seja, para conectar o sujeito a uma propriedade composta, usa-se um nodo branco como um nodo de ligação (Figura 2.3(b)). Em RDFS um nodo branco pode ser utilizado como “substituto”, para corrigir situações em que um literal pode aparecer na posição do sujeito, o que não é permitido pelo modelo RDF. Quando se utiliza SPARQL, a linguagem padrão para consultas em bases RDF, nodos brancos são utilizados na cláusula WHERE para representar variáveis que podem ser relacionadas aleatoriamente, mas não são retornadas na solução.

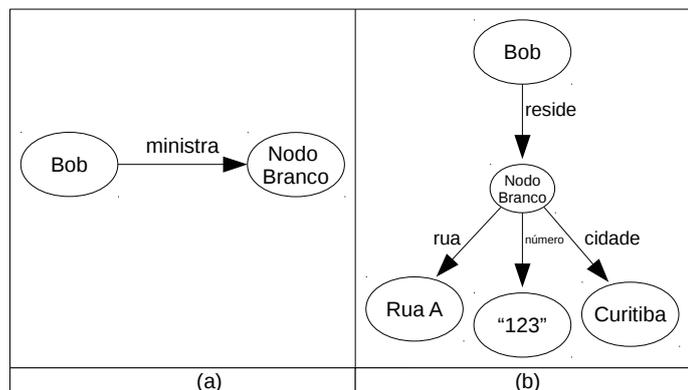


Figura 2.3: (a) Nodos brancos representando URI desconhecida. (b) *blank nodes* usado em propriedade n-ária.

A proposta de uso de nodos brancos neste trabalho não viola estas formas de utilização, mas propõe sua utilização na manutenção da consistência de uma base RDF sem a necessidade de operações em cascata. Esta proposta está de acordo com o contexto teórico no sentido de caracterizar uma informação existencial e fazer o papel de uma

informação nula em alguns casos. Esta proposta será descrita em maiores detalhes nos próximos capítulos.

2.3 DISCUSSÃO

Neste capítulo foram apresentados conceitos relacionados a RDF e nodos brancos. Em especial, a distinção entre esquema e instância de uma base RDF, que constitui uma parte importante deste trabalho e que não está presente na definição padrão do modelo RDF. Outro conceito destacado é a semântica de nodos brancos que originalmente não se relaciona com as restrições de integridade, mas que na estratégia desenvolvida nesta tese assume papel importante no processo. No próximo capítulo são tratados alguns trabalhos relacionados com estes temas e com a abordagem utilizada no sistema BNS.

3 TRABALHOS RELACIONADOS

Diversos trabalhos relatam a importância de garantir a integridade das bases de dados e, em particular de bases RDF [FKAC13], [AH06],[FPA⁺16]. As atualizações em bases RDF podem causar alterações no grafo RDF, o que evidencia seu potencial de causar efeitos indesejáveis. Assim, o controle de alterações requer esforços para garantir que a integridade de uma base de dados seja preservada [MSCK05].

Para o levantamento de trabalhos relacionados, além do contexto de controle de alterações, analisou-se ainda o aspecto do uso e interpretação de nodos brancos nas soluções adotadas. Embora, de modo geral, as soluções não utilizem nodos brancos com o mesmo objetivo adotado neste trabalho, é importante considerar como as soluções existentes tratam este tema para que fique clara a semântica aqui adotada e eventuais diferenças em relação a trabalhos encontrados na literatura.

A seguir são apresentados alguns trabalhos divididos em duas categorias: restrições de integridade e nodos brancos.

3.1 RESTRIÇÕES DE INTEGRIDADE

O controle das alterações é importante para garantir a consistência dos dados, principalmente considerando que atualmente há um grande volume de dados RDF [AH06]. A consistência deve levar em conta diversas restrições impostas sobre os dados. O primeiro tipo de restrição se refere ao modelo RDF, por exemplo a definição de classes, propriedades ou instâncias.

O segundo tipo se refere ao esquema RDFS. Ele leva em conta as propriedades relacionadas diretamente aos conceitos do modelo RDF, ou seja, restrições de domínio, imagem, subclasses e subpropriedades. Por exemplo, na inserção de uma tripla (s, p, o) é preciso verificar se a propriedade p liga um sujeito s , que é uma instância do domínio de p , a um objeto o que é uma instância da sua imagem.

Por fim, ainda é possível que restrições específicas de uma determinada base de dados, definidas pela aplicação, sejam impostas aos dados. Para exemplificar, uma base de dados acadêmica pode restringir que todo professor de uma instituição esteja associado a uma disciplina sob sua responsabilidade.

Alguns trabalhos na literatura se dedicam a atender apenas restrições RDF [AH06], restrições RDFS [MSCK05] e [VG06] ou ainda, além destas, restrições da aplicação [FKAC13].

Uma proposta centrada na ideia de versionamento de bases RDF, propõe uma abordagem baseada em sistemas CVS (*Concurrent Version System*) [VG06]. A estratégia é dividida em algumas etapas. Em primeiro lugar, é feita a identificação e tratamento de diferenças semânticas e estruturais entre as versões. Em seguida, faz-se a intercalação e identificação de conflitos entre as versões.

Outro trabalho nesta linha [FPA⁺16], cria uma relação entre cada instrução de inserção e/ou remoção de triplas e seu grafo correspondente e mantém as informações adicionais como o autor da instrução e o motivo da alteração solicitada. Estas informações permitem que seja possível desfazer alterações, ou mesmo realizar uma intercalação entre diferentes versões. Uma função *hash* é definida para identificar as alterações e agrupar em um mesmo valor as diferentes versões, facilitando a reversão se necessário.

Trabalhos com esta temática poderiam ser utilizados em conjunto com o sistema BNS, para que, além do processo de garantia de integridade implementado na ferramenta, as alterações pudessem, se necessário, ser desfeitas retornando assim ao estado consistente anterior à atualização.

Uma abordagem para controle de evolução de ontologias [AH06], merece atenção especial por ser o primeiro a considerar nodos brancos. A solução divide um grafo RDF em grafos atômicos, ou seja, a menor unidade que não pode ser dividida sem duplicar nodos brancos. Desta forma, os grafos atômicos são conectados por nodos brancos, ou seja não fazem parte de nenhum dos subgrafos. Após as alterações, os subgrafos podem ser reconectados aos nodos brancos para recompor o grafo novamente. No entanto, com esta abordagem uma eventual necessidade de reverter alterações fica dificultada, já que o trabalho propõe apenas uma forma de representar as mudanças, e não um processo de controle como um todo. Portanto, diferente do sistema BNS, que propõe o uso de nodos brancos na solução, neste trabalho os nodos brancos são isolados para evitar conflitos com versões diferentes do grafo.

Uma sequência deste trabalho [FES⁺16], considera a evolução de ontologias em ambientes com dados replicados. O trabalho propõe uma abordagem para sincronização e resolução de conflitos para tratar a propagação de alterações entre as réplicas. Diferentemente do trabalho anterior, neste há uma proposta de resolução de conflitos automatizada que decide sobre a política de resolução conforme o tipo do dado envolvido. A proposta apresentada nesta tese é toda automatizada, portanto sem interferência do usuário para resolver conflitos, mas restringe as operações de atualização somente sobre as instâncias da base.

A manutenção de integridade de uma base RDF pode ser definida através de uma linguagem de atualização declarativa (RUL)[MSCK05]. Cada possível alteração definida na linguagem tem um conjunto bem definido de efeitos colaterais, que são aplicados automaticamente pelo sistema, garantindo sua integridade. Para isto é necessário que existam definições de restrições de domínio e de imagem e que as classes e propriedades estejam todas definidas explicitamente na base.

O Sistema FKAC [FKAC13], propõe um framework para tratar de alterações em bases RDF. Em relação aos demais, ele tem como diferencial tratar de restrições da aplicação, além das restrições do modelo RDF e do esquema. O principal objetivo do trabalho está relacionado à garantia de integridade depois da execução de alterações na base.

O trabalho trata tanto de operações de atualização envolvendo instâncias, quanto o esquema. Além disso, ele é baseado no princípio da alteração mínima, ou seja, é preciso que a integridade da base de dados seja restabelecida com o menor número de operações possível. Assim, muitas vezes a execução de uma operação que viole a integridade pode ter mais de um conjunto de operações de efeitos colaterais. O sistema calcula todos eles e depois, baseado no peso atribuído para as operações, calcula uma pontuação final para cada solução, o que leva ao cálculo da diferença (delta) entre as soluções. A partir deste cálculo, opta-se pela solução de menor custo.

Outra proposta baseada no princípio de alteração mínima, trata remoções de dados em ontologias através da substituição por uma super classe ou subclasse do dado removido, eliminando assim a necessidade de remoções em cascata [SG13]. Este objetivo é similar ao sistema BNS. O processo é automatizado, mas pode ter interferência do usuário em casos nos quais este chamado dado “equivalente” não é encontrado.

Ainda na abordagem de manutenção de integridade, alguns trabalhos se dedicam à investigação de dependências em grafos. Um deles propõe uma classe de dependência para grafos denominada *graph entity dependencies* (GED) [FL17]. Ela expressa dependências funcionais em grafos e auxilia na identificação de inconsistências.

A ideia de validação de grafos também está presente em tecnologias tais como ShEx ([Sh14]) e SHACL ([KR17]). Embora o foco destes trabalhos sejam no esquema RDF e o do sistema BNS seja em restrições de integridade, um estudo sobre a interação de ambas as estratégias requer uma investigação no futuro. No entanto, nenhum deles trata a manutenção de consistência em função de atualizações.

Por fim, um outro trabalho similar também propõe a atualização de uma base de dados seguindo o princípio de alteração mínima mantendo a consistência em relação a restrições de integridade [HFL17]. No entanto, este trabalho utiliza efeitos colaterais em cascata, evitando-os apenas em casos nos quais é possível aplicar o princípio de transações. Ou seja, se uma restrição impõe que todo professor(p) ministra pelo menos uma disciplina (d), caso uma atualização remova a única disciplina ministrada por p , pelo efeito cascata p deve ser removido da base. Mas se na mesma transação é feita uma inserção de d_1 ministrada por p , então estas operações, implementadas de forma atômica, garantem a integridade da base sem remoções em cascata.

3.1.1 Análise Comparativa

A Tabela 3.1 apresenta uma síntese comparativa entre os trabalhos relacionados às restrições de integridade, resumindo o tipo de restrição tratada e qual o contexto no qual a solução apresentada foi proposta.

Trabalhos que abordam Restrições de Integridade		
Trabalhos	Tipo de Restrição	Solução Adotada
[FKAC13]	RDF, RDFS, Aplicação	Framework para tratamento de alterações
[MSCK05]	RDF, RDFS	Linguagem declarativa para atualização RDF
[FPA ⁺ 16]	RDF, RDFS	Versionamento de bases RDF
[AH06]	RDF	Evolução de ontologias
[VG06]	RDF e RDFS	Versionamento de bases RDF
[FES ⁺ 16]	RDF e RDFS	Evolução de ontologias
[SG13]	Aplicação	Adaptação automática de ontologias
[FL17]	Grafos	Identificação de dependências em grafos
[Sh14]	RDF	Linguagem para especificação formal de restrições
[KR17]	RDF, RDFS	Linguagem para validação de grafos RDF
[HFL17]	RDF, RDFS, Aplicação	Estratégia para atualização determinística

Table 3.1: Trabalhos Relacionados - Restrições de Integridade

3.2 NODOS BRANCOS

Além do contexto de controle de alterações, analisou-se ainda o aspecto do uso e interpretação de nodos brancos nas soluções adotadas. Embora as soluções não utilizem nodos brancos para auxiliar no controle de alterações, diversos trabalhos fazem considerações sobre o tema. Por exemplo, duas versões de uma base de dados (antes e depois de uma atualização) podem ter nodos brancos que podem ser unificados, o que significa dizer que trata-se do mesmo nodo branco nas duas versões. Como a identificação de um nodo branco não tem uma URI com identificação única, conforme explicado na Seção 2.1, detectar esta unicidade não é um processo simples. Vários trabalhos apresentados na Seção 3.1 consideram nodos brancos em suas soluções.

O primeiro trabalho [AH06] a tratar alterações em grafos com nodos brancos, apresenta o conceito de grafo atômico, como sendo a menor unidade que não pode ser dividida sem duplicar nodos brancos. Assim, é feita uma divisão de modo que os subgrafos sejam conectados por um nodo branco e desta forma as alterações podem ser feitas em subgrafos sem preocupação com alterações em nodos brancos. Após as alterações os subgrafos são novamente conectados. Conflitos que possam surgir a partir de alterações solicitadas são resolvidas com a interferência do usuário.

Existem trabalhos que tratam do “enriquecimento” de nodos brancos [VG06], no qual informações são adicionadas a um nodo branco e utilizadas para identificar nodos iguais que podem estar replicados em diferentes versões.

Por fim, uma técnica chamada de MSG (*Minimum Self-Contained Graph*) [FPA⁺16], propõe armazenar cada nodo branco com um conjunto de triplas contendo este nodo branco. Assim é possível controlar inserções e remoções de triplas contendo nodos brancos e manter a identificação correta do nodo.

Uma outra estratégia se dedica a identificar nodos brancos iguais com o objetivo final de remover aqueles que são redundantes de uma base RDF [YHLX18]. Isto é feito a partir da identificação de sujeitos e objetos ligados a um nodo branco e o agrupamento dos dados similares.

Uma outra abordagem que considera a existência de nodos brancos é a skolemização. Neste processo é possível substituir uma variável y quantificada existencialmente, por um novo termo $f(x_1, x_2, \dots, x_n)$. Este termo f é definido como uma função de Skolem. Assim, $\forall x \exists y R(x, y) \Leftrightarrow \exists f \forall x, R(x, f(x))$. No entanto, quando a variável que está sendo substituída não depende de outro termo, como no caso de RDF [Hog15], esta função pode ser substituída por uma constante. Esta constante é utilizada como uma identificação para o nodo branco, uma vez que eles não possuem identificação através de URI, conforme apresentado na seção 2.2.

Diversos trabalhos tratam a skolemização para criação de identificadores de nodos brancos [Hog15] [Mn16]. Estes identificadores auxiliam no reconhecimento de nós idênticos em grafos distintos e assim permitir a identificação de grafos isomorfos [Hog15].

3.2.1 Análise Comparativa

É primordial tratar nodos brancos e a incompletude de sua definição quando se aborda versionamento de bases RDF. Assim, boa parte dos trabalhos tratados anteriormente neste tema, também consideram nodos brancos em suas soluções. Além disso, o tema *skolemização* também é estudado quando se trata de nodos brancos. A tabela 3.2 sumariza os trabalhos que tratam de nodos brancos, relacionando a proposta e o objetivo a que se propõe os trabalhos.

Trabalhos que abordam nodos brancos		
Trabalhos	Objetivo do Trabalho	Semântica de nodos brancos
[FPA ⁺ 16]	Versionamento de bases RDF	Variáveis existenciais
[AH06]	Evolução de ontologias	Variáveis existenciais
[VG06]	Versionamento de bases RDF	Variáveis existenciais
[Mn16]	Manutenção de Integridade	Skolemização
[Hog15]	Isomorfismo de grafos RDF	Skolemização
[YHLX18]	Remoção de Nodos Brancos Redundantes	Variáveis existenciais

Table 3.2: Trabalhos Relacionados - Nodos Brancos

3.3 DISCUSSÃO

Os trabalhos apresentados neste capítulo estão divididos em duas categorias por questão de organização. Como pode ser observado, existem várias interseções e alguns trabalhos aparecem relacionados a ambos os temas. No entanto, o objetivo principal é, em primeiro lugar, identificar temas abordados na tese e, por fim, diferenciar as abordagens utilizadas em outros trabalhos da estratégia proposta neste trabalho.

É possível observar como o tema “Nodos Brancos” é tratado de forma recorrente em trabalhos relacionados a bases RDF. No entanto, a ideia de utilizá-los para atender restrições de integridade é algo novo e que se propõe a resolver consequências indesejáveis como as operações em cascata ou a utilização de operações mais complexas como a fragmentação de grafos para se obter a consistência da base.

No próximo capítulo são apresentados maiores detalhes sobre as restrições de integridade consideradas nesta tese e como minimizar as operações em cascata com o uso de nodos brancos.

4 RESTRIÇÕES DE INTEGRIDADE E EFEITOS COLATERAIS

Neste capítulo são apresentadas as restrições de integridade e outros conceitos utilizados nesta tese. Para manter a consistência da base de dados após as atualizações, é preciso determinar o conjunto de efeitos colaterais, ou seja, as atualizações adicionais necessárias para satisfazer as restrições de integridade. Para isto, são consideradas apenas atualizações de instância.

4.1 RESTRIÇÕES DE INTEGRIDADE

Inicialmente é preciso apresentar algumas notações e definições utilizadas ao longo da tese. Conforme definido no Capítulo 2, INSTPRED é o conjunto de predicados referentes à instância da base, enquanto SCHPRED são os predicados do esquema da base. Desta forma, a primeira definição refere-se à base de dados.

Definição 4.1 (Base de Dados) *Uma base de dados RDF é uma tripla $\Delta = (D, D_{Sch}, \Sigma)$, onde D é uma instância da base de dados (um conjunto de fatos com predicados em INSTPRED), D_{Sch} é o esquema da base de dados (um conjunto de fatos com predicados em SCHPRED) e $\Sigma = (\mathcal{A}, \mathcal{C})$ é um conjunto de restrições, onde \mathcal{A} é o conjunto de restrições do esquema RDF e \mathcal{C} é o conjunto de restrições da aplicação. \square*

Considere que $var = \{X_1, X_2, \dots, Y_1, \dots\}$ é um conjunto infinito de variáveis sobre valores em $\mathbf{A}_C \cup \mathbf{A}_N$ (conforme definido no Capítulo 2, \mathbf{A}_C é o conjunto de constantes e \mathbf{A}_N é o conjunto de nodos brancos). X é usado como uma abreviação para $X_1 \dots X_k$, onde $k \geq 0$. Uma fórmula atômica (ou simplesmente átomo) tem a forma $P(t_1, \dots, t_n)$, onde P é um predicado n -ário, e t_1, \dots, t_n são termos.

As restrições do esquema RDF são baseadas nas regras do esquema RDF (RDFS). Como exemplo de uma restrição, pode-se definir que se uma entidade é uma instância de uma classe então ela é também instância de suas superclasses, da seguinte forma: $r = CI(X_1, X_2) \wedge CSub(X_2, X_3) \rightarrow CI(X_1, X_3)$. O lado esquerdo da regra é chamado de corpo da regra ($corpo(r)$), enquanto o lado direito é chamado de cabeça da regra ($cabeça(r)$).

As restrições da aplicação são regras definidas baseadas na semântica específica de cada aplicação.

Definição 4.2 (Restrições da Aplicação) *Considere que c_1, c_2 são rótulos de classes, e p_1, p_2 são rótulos de propriedades em \mathbf{A}_C . Um conjunto \mathcal{C} de restrições da aplicação é composto de regras nos formatos apresentados na Tabela 4.1.*

Tipo 1:	$CI(X_1, c_1) \rightarrow CI(X_1, c_2)$ $PI(X_1, X_2, p_1) \rightarrow PI(X_1, X_2, p_2)$	$CI(X_1, c_1) \rightarrow \neg CI(X_1, c_2)$ $PI(X_1, X_2, p_1) \rightarrow \neg PI(X_1, X_2, p_2)$
Tipo 2:	$CI(X_1, c_1) \rightarrow PI(X_1, X_2, p_1)$ $CI(X_2, c_1) \rightarrow PI(X_1, X_2, p_1)$	$CI(X_1, c_1) \rightarrow \neg PI(X_1, X_2, p_1)$ $CI(X_2, c_1) \rightarrow \neg PI(X_1, X_2, p_1)$
Tipo 3:	$PI(X_1, X_2, p_1) \rightarrow CI(X_1, c_1)$ $PI(X_1, X_2, p_1) \rightarrow CI(X_2, c_1)$	$PI(X_1, X_2, p_1) \rightarrow \neg CI(X_1, c_1)$ $PI(X_1, X_2, p_1) \rightarrow \neg CI(X_2, c_1)$

Table 4.1: Tipos de Restrições de Integridade

São impostas as seguintes restrições sobre \mathcal{C} : (1) se r_1 é uma restrição do Tipo 2, então não existe nenhuma restrição $r_2 \in \mathcal{C}$ tal que $\text{cabeca}(r_1)$ e $\text{corpo}(r_2)$ sejam unificáveis¹; (2) se r_1 é uma restrição do Tipo 3, então não existe nenhuma restrição $r_2 \in \mathcal{C}$ tal que $\text{corpo}(r_1)$ e $\text{cabeca}(r_2)$ sejam unificáveis. \square

Embora nesta tese exista a restrição de que as variáveis no corpo apareçam na cabeça para regras do Tipo 1 e 3, é feita uma flexibilização, permitindo um átomo negativo na cabeça da regra. Na Seção 4.3, é mostrado que efeitos colaterais de regras do tipo 2 podem gerar novos nodos brancos. Neste trabalho, foram impostas estas limitações sobre as restrições para evitar a propagação de nodos brancos. Assim, evitam-se problemas relacionados ao *chase* infinito [CGK08]. O próximo exemplo apresenta restrições que respeitam estas limitações.

Exemplo 4.1 Considere um conjunto de restrições \mathcal{C} definido para uma aplicação acadêmica. Elas definem que apenas pesquisadores podem coordenar projetos (r_1) e eles devem também ser membros deste projeto(r_3). Além disso, todos os pesquisadores devem ser membros de no mínimo um projeto(r_2). Pesquisadores são professores(r_4), e professores não podem ser estudantes(r_5). Por fim, professores precisam ministrar pelo menos uma disciplina(r_6).

$$\begin{aligned} r_1 &: PI(X_1, X_2, \text{coordena}) \rightarrow CI(X_1, \text{Pesquisador}) \\ r_2 &: CI(X_1, \text{Pesquisador}) \rightarrow PI(X_1, X_2, \text{membro}) \\ r_3 &: PI(X_1, X_2, \text{coordena}) \rightarrow PI(X_1, X_2, \text{membro}) \\ r_4 &: CI(X_1, \text{Pesquisador}) \rightarrow CI(X_1, \text{Professor}) \\ r_5 &: CI(X_1, \text{Professor}) \rightarrow \neg CI(X_1, \text{Estudante}) \\ r_6 &: CI(X_1, \text{Professor}) \rightarrow PI(X_1, X_2, \text{ministra}) \end{aligned}$$

4.2 ATUALIZAÇÕES

Uma operação de inserção é denotada por um átomo positivo, enquanto um átomo negativo denota uma remoção. A semântica destas operações é direta para inserção e remoção de entidades. Assim, a partir de uma base de dados D , a inserção de uma instância a de uma determinada classe c , $CI(a, c)$, resulta em $(D \cup \{CI(a, c)\})$, e a remoção de um nodo branco N_1 , $\neg BN(N_1)$, resulta em $(D \setminus \{BN(N_1)\})$. No entanto, a inserção e remoção de propriedades podem conter o caracter especial “_” como termo. Estas operações possuem um significado especial, conforme a definição na Tabela 4.2. As operações que não estão listadas na tabela não tem efeito sobre a base de dados.

Para cada operação, é mostrado o conjunto de operações a serem adicionadas (átomos positivos) e removidas (átomos negativos) da base de dados D . Intuitivamente, nodos brancos são inseridos em uma base de dados D apenas se não existir nenhuma outra entidade em D que desempenhe o mesmo papel. Portanto, uma inserção de uma propriedade p sem nodos brancos ($PI(a, b, p)$) substitui uma propriedade que envolva nodos brancos (e portanto gera $\neg PI(a, N_1, p)$ e $\neg PI(N_1, b, p)$) que eventualmente já existam na base de dados. Ou seja, é adotada a semântica padrão de nodos brancos como variáveis existenciais, desempenhando o papel de algum recurso desconhecido. Desta forma, quando uma entidade é atribuída para desempenhar o mesmo papel que o nodo branco, o recurso passa a ser conhecido e portanto o nodo branco pode ser removido.

¹ $\text{cabeca}(r_1)$ e $\text{corpo}(r_2)$ são unificáveis se existe uma substituição de variáveis por variáveis ou constantes que torne os átomos idênticos

Table 4.2: Semântica das Operações de Atualização

$\llbracket CI(a, c) \rrbracket_D =$	$\{CI(a, c)\}$	$\llbracket \neg CI(a, c) \rrbracket_D =$	$\{\neg CI(a, c)\}$
$\llbracket \neg Ind(a) \rrbracket_D =$	$\{\neg Ind(a)\} \cup$ $\{\neg PI(X, a, p) \mid PI(X, a, p) \in D\} \cup$ $\{\neg PI(a, X, p) \mid PI(a, X, p) \in D\} \cup$ $\{\neg CI(a, X) \mid CI(a, X) \in D\}$	$\llbracket Ind(a) \rrbracket_D =$	$\{Ind(a)\}$
$\llbracket BN(N_1) \rrbracket_D =$	$\{BN(N_1)\}$	$\llbracket \neg BN(N_1) \rrbracket_D =$	$\{\neg BN(N_1)\}$
$\llbracket PI(a, b, p) \rrbracket_D =$	$\{PI(a, b, p)\} \cup$ $\{\neg PI(N_1, b, p), \neg BN(N_1) \mid PI(N_1, b, p) \in D\} \cup$ $\{\neg PI(a, N_1, p), \neg BN(N_1) \mid PI(a, N_1, p) \in D\}$	$\llbracket \neg PI(a, b, p) \rrbracket_D =$	$\{\neg PI(a, b, p)\}$
$\llbracket PI(_, b, p) \rrbracket_D =$	se existe X tal que $PI(X, b, p) \in D$ então $\{$ senão $\{PI(N_1, b, p), BN(N_1)\}$	$\llbracket \neg PI(_, b, p) \rrbracket_D =$	$\{\neg PI(X, b, p) \mid$ $PI(X, b, p) \in D\}$
$\llbracket PI(a, _, p) \rrbracket_D =$	se existe X tal que $PI(a, X, p) \in D$ então $\{$ senão $\{PI(a, N_1, p), BN(N_1)\}$	$\llbracket \neg PI(a, _, p) \rrbracket_D =$	$\{\neg PI(a, X, p) \mid$ $PI(a, X, p) \in D\}$
$\llbracket \neg PI(N_1, b, p) \rrbracket_D =$	$\{\neg PI(N_1, b, p)\}$	$\llbracket \neg PI(a, N_1, p) \rrbracket_D =$	$\{\neg PI(a, N_1, p)\}$

Para representar a inserção condicional de um nodo branco, é introduzido o caracter especial “ $_$ ” em operações de inserção. Assim, a inserção $PI(a, _, p)$ ou $PI(_, a, p)$ tem efeito no banco de dados somente se a entidade a não estiver associada a nenhuma outra com a propriedade p . Neste caso é gerado na base de dados $PI(a, N_1, p)$ ou $PI(N_1, a, p)$, além de $BN(N_1)$. Em operações de remoção de propriedades, o caracter especial “ $_$ ” representa a remoção de todas as associações p de uma entidade. Assim, a operação $\neg PI(a, _, p)$ ou $\neg PI(_, a, p)$ tem como consequência a remoção de todas as propriedades p associadas à entidade a . Denota-se por $D \uplus I$ o resultado da aplicação do conjunto de operações de atualização I sobre a base de dados D . Assim, para um fato positivo f , se $f \in D$ então $D \uplus \{f\} = D$. Para um fato negativo $\neg f$ se $f \notin D$ então $D \uplus \{\neg f\} = D$. Baseado na semântica das operações, pode-se definir a noção de subordinação entre operações.

Definição 4.3 (Subordinação de Atualizações) *A operação op_1 é subordinada por uma operação op_2 , denotada por $op_1 \preceq op_2$ se para qualquer base de dados D , $(\llbracket op_1 \rrbracket_D \cup \llbracket op_2 \rrbracket_D) = \llbracket op_2 \rrbracket_D$. Ou seja, o efeito, sobre a base de dados, de executar op_1 e op_2 é o mesmo de executar apenas op_2 . Dado o conjunto de operações de atualização upd , denota-se por $clean(upd)$ um subconjunto de upd sem operações subordinadas. Ou seja, se op_1 e $op_2 \in upd$ e $op_1 \preceq op_2$, então $clean(upd)$ não contém op_1 .*

Como um exemplo, se $PI(Bob, _, coordena) \preceq PI(Bob, ProjDB, coordena)$, então após adicionar $PI(Bob, ProjDB, coordena)$ a operação $PI(Bob, _, coordena)$ não tem mais efeito sobre a base de dados. Ou ainda, $\neg PI(Bob, ProjDB, coordena) \preceq \neg PI(Bob, _, coordena)$, uma vez que $\neg PI(Bob, _, coordena)$ exige a remoção de todas as propriedades $coordena$ de Bob , inclusive $PI(Bob, ProjDB, coordena)$.

Outra questão importante é definir quando um conjunto de atualizações é consistente.

Definição 4.4 (Conjunto de Atualizações Consistente) *Um conjunto de atualizações I é consistente se não há dois átomos op_1 e op_2 tais que $op_1 = \neg op_2$ ou op_2 é da forma $\neg \overline{op_2}$, onde $\overline{op_2} = h(op_1)$ ou $h(\overline{op_2}) = op_1$, e h é um homomorfismo de $(\mathbf{A}_N \cup \mathbf{A}_C \cup \{_\}) \rightarrow (\mathbf{A}_N \cup \mathbf{A}_C \cup \{_\})$. \square*

O exemplo 4.2 mostra diferentes conjuntos de atualização inconsistentes de acordo com a Definição 4.4.

Exemplo 4.2 *Considere diferentes conjuntos de atualização:*

$$I_1 = \{\neg PI(Bob, BD, ministra), PI(Bob, _, ministra)\}$$

$$I_2 = \{PI(Bob, BD, ministra), \neg PI(Bob, _, ministra)\}$$

$$I_3 = \{PI(Bob, _, ministra), \neg PI(Bob, _, ministra)\}$$

$$I_4 = \{PI(Bob, BD, ministra), \neg PI(Bob, BD, ministra)\}$$

As atualizações I_2, I_3 e I_4 são inconsistentes como consequência direta da semântica das operações. No caso da atualização I_1 a inconsistência se deve a um detalhe mais sutil. Ao remover uma relação *ministra* de *Bob* e ao mesmo tempo inserir uma relação com nulo, o sistema deveria ter a capacidade de identificar que esta entidade desconhecida (caracter especial $_$) não pode ser o objeto *BD*, já que esta relação está sendo removida. No entanto, o modelo adotado nesta tese não permite a representação que um nodo nulo não pode ser substituído por uma determinada entidade ou literal. Assim, esta atualização é considerada inconsistente.

4.3 EFEITOS COLATERAIS BASEADOS EM RESTRIÇÕES DA APLICAÇÃO

Para computar os efeitos colaterais das restrições da aplicação, derivados de um conjunto de atualizações I , inicialmente são apresentadas algumas notações. Elas denotam subconjuntos de I contendo operações de inserção e remoção, bem como operações envolvendo o caracter especial “ $_$ ”.

- $I = I^+ \cup I^-$ onde I^+ é o conjunto de átomos positivos e I^- é o conjunto de átomos negativos.
- $I = \bullet(I) \cup \ominus(I)$ onde $\bullet(I)$ é o conjunto de átomos (sem o caracter especial “ $_$ ”) e $\ominus(I)$ é o conjunto de átomos contendo o caracter especial “ $_$ ”.
- Consequentemente, $I^+ = \bullet(I^+) \cup \ominus(I^+)$ e $I^- = \bullet(I^-) \cup \ominus(I^-)$

Dados um conjunto de atualizações I e um conjunto de restrições da aplicação \mathcal{C} (definido na seção 4.1), são introduzidas três operações usadas para computar efeitos colaterais. O primeiro operador (T) computa os efeitos colaterais seguindo a regra da esquerda para a direita, ou seja, do corpo para a cabeça. Portanto, se há uma inserção que casa com o corpo de uma regra r então deve-se adicionar atualizações para a cabeça da regra para satisfazer r . Como um exemplo, considere a regra $r_3 : PI(X_1, X_2, coordena) \rightarrow PI(X_1, X_2, membro)$ no Exemplo 4.1. Se $I = \{PI(John, projDB, coordena)\}$ então $T(I)$ contém $PI(John, projDB, membro)$. É preciso também percorrer as regras para trás, que é representado pelo operador ϑ . Continuando o mesmo exemplo, considere agora a remoção de $PI(John, projDB, membro)$, ou seja, $I = \{\neg PI(John, projDB, membro)\}$. Neste caso, fatos que exigem que $PI(John, projDB, membro)$ seja verdadeiro devem ser removidos da base de dados. Portanto, $\vartheta(I)$ contém $\neg PI(John, projDB, coordena)$.

Definição 4.5 (Operadores T e ϑ) Considere que $T_{\mathcal{C}}$ e $\vartheta_{\mathcal{C}}$ sejam operadores sobre um conjunto de restrições da aplicação \mathcal{C} e um conjunto de átomos I . Quando o conjunto \mathcal{C} é claro pelo contexto, escreve-se T (respectivamente, ϑ) ao invés de $T_{\mathcal{C}}$ (respectivamente, $\vartheta_{\mathcal{C}}$).

$T(I) = I \cup \{l \mid \exists c \in \mathcal{C} \wedge \text{há um homomorfismo } h \text{ tal que } h(\text{body}(c)) \in I \wedge l = \widehat{h}(\text{cabeca}(c)) \text{ onde } \widehat{h} \supseteq h \text{ é uma extensão de } h \text{ tal que quando } c \text{ é da forma } L_1(X) \rightarrow L_2(X, Y), \text{ então para todo } y_i \in Y, \text{ se } y_i \in \text{var}, \text{ então } \widehat{h}(y_i) = \text{"_"}.\}$

$\vartheta(I) = I \cup \{l \mid \exists c \in \mathcal{C} \wedge \text{há um homomorfismo } h \text{ tal que } h(\text{cabeca}(c)) \in I \wedge l = \widehat{h}(\text{corpo}(c)) \text{ onde } \widehat{h} \supseteq h \text{ é uma extensão de } h \text{ tal que quando } c \text{ é da forma } L_1(X, Y) \rightarrow L_2(X), \text{ para todo } y_i \in Y, \text{ se } y_i \in \text{var} \text{ então } \widehat{h}(y_i) = \text{"_"}.\} \square$

Quando \mathcal{C} tem apenas regras do Tipo 1 (Definição 4.2), operadores T e ϑ correspondem aos apresentados em [HLS98]. Este trabalho estende esta definição por conter átomos instanciados com o caracter especial “_”. Como as restrições tem formato especial, as boas propriedades anunciadas em [HLS98] ainda se mantém. Em particular, operadores T e ϑ são monotônicos e tem um *ponto fixo mínimo*. Denota-se por $T^*(I)$ e por $\vartheta^*(I)$, o ponto fixo mínimo de T e ϑ , respectivamente, sobre um conjunto de átomos I . Uma vez que todas as restrições em \mathcal{C} possuem átomos positivos como corpo, tem-se que $T^*(I^-) = I^-$ e $T^*(I^-)^- = \vartheta^*(I^-)^- = I^-$.

Exemplo 4.3 Considere o conjunto de restrições \mathcal{C} no Exemplo 4.1.

Considere a atualização $I = \{PI(\text{John}, \text{projDB}, \text{coordena})\}$.

Aplicando a Definição 4.5, obtém-se $T(I) = \{PI(\text{John}, \text{projDB}, \text{coordena}), CI(\text{John}, \text{Pesquisador}), PI(\text{John}, _, \text{membro}), PI(\text{John}, \text{projDB}, \text{membro}), CI(\text{John}, \text{Professor}), PI(\text{John}, _, \text{ministra}), \neg CI(\text{John}, \text{Aluno})\}$.

Agora considere $I = \{PI(\text{John}, \text{projDB}, \text{membro})\}$.

Aplicando a Definição 4.5, obtém-se $\vartheta(I) = \{PI(\text{John}, \text{projDB}, \text{membro}), CI(\text{John}, \text{Pesquisador}), PI(\text{John}, \text{projDB}, \text{coordena}), PI(\text{John}, _, \text{coordena})\}$.

A aplicação do operador ϑ pode estar restrita a alguns tipos de regras. Denota-se por $\vartheta|_{ty_{1,3}}$ a aplicação do operador ϑ apenas sobre as restrições do Tipo 1 e 3. Além dos operadores T e ϑ , é necessário um terceiro, η , que se aplica apenas a regras do Tipo 2. Para apresentar a ideia por trás da definição deste operador, considere a regra r_6 no Exemplo 4.1: $CI(X_1, \text{Professor}) \rightarrow PI(X_1, X_2, \text{ministra})$ e a operação de remoção $I = \{\neg PI(\text{John}, \text{DB}, \text{ministra})\}$. Um possível efeito colateral é remover $CI(\text{John}, \text{Professor})$ se DB é a única disciplina que John ministra. No entanto, esta remoção pode na sequência, exigir outras remoções como efeito colateral, como a remoção de $CI(\text{John}, \text{Pesquisador})$, de acordo com a regra r_4 , e a associação de John com os projetos que ele coordena, conforme a regra r_1 . Para evitar estas remoções em cascata, a aplicação do operador η inclui o caracter especial “_” como substituto para a disciplina exigida para John . Isto é, $\neg PI(\text{John}, \text{DB}, \text{ministra})$ inclui $PI(\text{John}, _, \text{ministra})$ como um efeito colateral, onde “_” é um substituto para um valor desconhecido.

Definição 4.6 (Operador η) O operador η (ou $\eta_{\mathcal{C}}$) é aplicado apenas a regras do Tipo 2. Considerando que I seja um conjunto de fatos:

$\eta(I) = \{l \mid \exists c \in \mathcal{C} \text{ tal que } c \text{ é do tipo 2} \wedge \text{há um homomorfismo } h \text{ tal que } h(\text{cabeca}(c)) \in I \wedge l = h_2(\text{cabeca}(c)) \text{ onde } h_2 \text{ é um homomorfismo tal que } h_2(X) = h(X) \text{ e para todo } y_i \in Y \text{ se } y_i \in \text{var} \text{ então } h_2(y_i) = \text{"_"}.\} \square$

Exemplo 4.4 Considere \mathcal{C} no Exemplo 4.1. Se $I = \{PI(\text{John}, \text{DB}, \text{ministra}), PI(\text{John}, \text{projDB}, \text{membro})\}$ então, em função de r_6 e r_2 , $\eta(I)$ contém $PI(\text{John}, _, \text{ministra})$, e $PI(\text{John}, _, \text{membro})$. \square

4.4 CONSISTÊNCIA DAS RESTRIÇÕES DA APLICAÇÃO

Uma vez que regras em \mathcal{C} podem ter átomos negativos ou positivos em suas cabeças, é importante determinar quando um conjunto de restrições \mathcal{C} é consistente, ou seja, se ele gera um conjunto consistente de operações de atualização como efeito colateral [HLS98].

Definição 4.7 (Conjunto consistente de restrições de aplicação) *Um conjunto de restrições de aplicação \mathcal{C} é consistente se para todo átomo l $(T^*(l))^+ \cap \neg.(T^*(l))^- = \emptyset$. \square*

Nesta tese, considera-se que o conjunto de atualizações dado como entrada é um conjunto de operações $upd = \{op_1, \dots, op_k, \neg op_{k+1}, \dots, \neg op_l\}$, onde cada op_i é um átomo positivo ou negativo, sem nodos brancos e sem o caracter especial “_”. Ou seja, operações envolvendo estes termos são gerados apenas como efeitos colaterais da aplicação de uma regra.

Uma vez que não apenas as operações em upd podem ser inconsistentes, mas também as operações geradas como efeito colateral, deve-se computar os efeitos colaterais de upd com relação a \mathcal{C} antes de executar as atualizações. Em outras palavras, o interesse é obter um conjunto de átomos U tal que: (i) $upd \subseteq U$; (ii) U é o menor conjunto contendo todas os efeitos colaterais impostos por \mathcal{C} sobre upd e (iii) para cada átomo positivo $l_0 \in U$ não há átomo negativo $\neg.l_1 \in U$ tal que $l_0 = h(l_1)$ onde h é um homomorfismo que substitui o caracter especial “_” por constantes. Por exemplo, U pode conter os átomos $\neg.PI(a, b, A)$ e $PI(a, _, A)$, mas não $PI(a, b, A)$ e $\neg.PI(a, _, A)$. Portanto, a noção de um *conjunto consistente de efeitos colaterais* U é diferente da noção de um conjunto consistente de atualizações, como na Definição 4.4. É importante observar que dado um conjunto de atualizações upd , a meta é computar os efeitos colaterais de \mathcal{C} *sem consultar a base de dados* – apenas o conjunto de atualizações solicitado e \mathcal{C} devem ser usados neste processo.

A consistência apresentada pela Definição 4.4 é imposta ao conjunto de atualizações de entrada upd e também aos efeitos colaterais gerados de forma imediata (ou seja, através da aplicação do operador T). No entanto, a noção de um conjunto de efeitos colaterais consistentes U (ou seja, após a aplicação dos operadores ϑ e η) é mais relaxada do que aquela estabelecida pela Definição 4.4.

Definição 4.8 (Consistência do Conjunto de Efeitos Colaterais) *Dado um conjunto de atualizações U , U é um conjunto consistente de efeitos colaterais se para cada átomo positivo $l_0 \in U$ não existe nenhum átomo negativo $\neg.l_1 \in I$ tal que $l_0 = h(l_1)$ onde h é um homomorfismo de $(\mathbf{A}_N \cup \{_\}) \rightarrow (\mathbf{A}_N \cup \mathbf{A}_C \cup \{_\})$. \square*

Retornando ao Exemplo 4.2, pela Definição 4.4 todos os exemplos são conjuntos de atualização inconsistentes. No entanto, de acordo com a Definição 4.8, o conjunto I_1 é consistente, enquanto os conjuntos I_2 , I_3 , e I_4 são inconsistentes. De fato, I_2 , I_3 , e I_4 são consequências diretas da semântica das operações. A discussão sobre a consistência de I_1 é mais sutil. Primeiro, considere que se $PI(Bob, _, ministra)$ está em I_1 deve existir uma regra que impõe a sua inserção a partir por exemplo de $CI(Bob, Professor)$. Por outro lado $\neg PI(Bob, BD, ministra)$ também está em I_1 . Na semântica tradicional, esta atualização exigiria a remoção de $CI(Bob, Professor)$ de acordo com a regra r_6 . Mas isto é inconsistente com a inserção de $PI(Bob, _, ministra)$. Intuitivamente, não é clara qual seria a intenção do usuário neste caso. Uma vez que a Definição 4.4 está focada na consistência do conjunto de atualizações, esta solicitação de atualização é considerada inconsistente. O processo de computar efeitos colaterais continua para conjuntos de atualização consistentes com a aplicação dos operadores ϑ e η . A Definição 4.8 se aplica ao

resultado deste processo. Considere novamente a regra r_6 do exemplo 4.1 e uma solicitação de atualização $\neg PI(Bob, BD, ministra)$. A aplicação do operador η gera como efeito colateral $PI(Bob, _, ministra)$ como técnica para interromper atualizações em cascata. Portanto, conforme a Definição 4.8, o conjunto I_1 é consistente porque ele é formado a partir de um conjunto de atualização consistente (e portanto sem a ambiguidade mencionada anteriormente) e a existência de $\neg PI(Bob, BD, ministra)$ e $PI(Bob, _, ministra)$ deve ser resultado da aplicação do operador η .

4.5 EFEITOS COLATERAIS BASEADOS EM RESTRIÇÕES DO ESQUEMA RDF

Nesta seção são tratados os efeitos colaterais baseados em restrições do esquema RDF. Como definido na Seção 2.1, um esquema de banco de dados D_{Sch} é o conjunto de fatos com predicados em $SCHPRED = \{Cl, Pr, CSub, Psub, Dom, Rng\}$.

Exemplo 4.5 Em um sistema acadêmico, pode-se definir, por exemplo, que existem as classes *Projeto*, *Disciplina* e *Pessoa*, e que *Professor* e *Aluno* são subclasses de *Pessoa*. Também é possível definir que o domínio da propriedade *membro* é *Pessoa*, e que sua imagem é *Projeto*, da seguinte forma:

$$\begin{array}{lll}
 s_1 : Cl(Projeto) & s_4 : Cl(Professor) & s_7 : CSub(Professor, Pessoa) \\
 s_2 : Cl(Disciplina) & s_5 : Cl(Aluno) & s_8 : CSub(Aluno, Pessoa) \\
 s_3 : Cl(Pessoa) & s_6 : Pr(membro) & s_9 : Dom(membro, Pessoa) \\
 & & s_{10} : Rng(membro, Projeto)
 \end{array}$$

Define-se um conjunto de restrições do esquema RDF que deve ser satisfeito por *todas* as instâncias da base de dados. De forma similar à ideia de restrições de aplicação, as restrições de esquema são usadas para derivar fatos adicionais para completar a base de dados com todos os fatos conhecidos e manter a base de dados consistente com relação ao esquema D_{Sch} . A Tabela 4.3 mostra o conjunto de restrições proposto por [FKAC13]. A Tabela 4.4 apresenta as regras que foram modificadas para considerar a existência de nodos brancos. Desta forma, nodos brancos são distintos de propriedades, entidades e classes (regras b_1 - b_3). Eles são usados ainda para interromper a criação de nodos brancos em cascata. Isto é feito permitindo que propriedades se conectem a nodos brancos mesmo quando há um tipo definido para o sujeito e o objeto (regras b_6 e b_7). Como um exemplo, considere a inserção do fato $PI(N_1, projDB, membro)$. De acordo com o esquema do banco de dados definido no Exemplo 4.5, o domínio de *membro* é *Pessoa*, mas a regra b_6 é satisfeita. Em [FKAC13], a regra b_6 foi originalmente definida como $PI(x, y, z) \wedge Dom(z, w) \rightarrow CI(x, w)$. Ou seja, para satisfazer a restrição o seguinte fato deveria ser inserido na base de dados: $CI(N_1, Pessoa)$. Neste caso, o nodo branco deveria ser “tipado” e todas as propriedades exigidas para *Pessoa* seriam exigidas também para o nodo branco N_1 , o que poderia exigir a geração de um número infinito de nodos brancos.

4.6 DISCUSSÃO

Em relação às restrições de integridade, foram destacados os três tipos de restrições de aplicação utilizados no escopo deste trabalho e a semântica das operações de atualização que foi definida na Tabela 4.2.

Também foram destacadas as definições de efeitos colaterais para regras de aplicação e regras do esquema RDF para compreensão de como é tratado o mecanismo de

Table 4.3: Subconjunto de restrições de esquema [FKAC13]

m_1 :	$Cl(x) \wedge Pr(y) \rightarrow (x \neq y)$
m_2 :	$Cl(x) \rightarrow Csub(x, rdfs:Resource)$
m_3 :	$Ind(x) \rightarrow CI(x, rdfs:Resource)$
m_4 :	$Csub(x,y) \wedge Csub(y,z) \rightarrow Csub(x,z)$
m_5 :	$Pr(x) \rightarrow Dom(x,y) \wedge Rng(x,z)$
m_6 :	$CI(x,y) \rightarrow Ind(x)$
m_7 :	$CI(x,y) \rightarrow Cl(y) \vee (y=rdfs:Resource)$
m_8 :	$PI(x,y,z) \rightarrow Pr(z)$
m_9 :	$CI(x,y) \wedge Csub(y,z) \rightarrow CI(x,z)$
m_{10} :	$PI(x,y,z) \wedge Psub(z,w) \rightarrow PI(x,y,w)$

Table 4.4: Restrições que envolvem nodos brancos

b_1 :	$Pr(x) \wedge BN(y) \rightarrow (x \neq y)$
b_2 :	$Ind(x) \wedge BN(y) \rightarrow (x \neq y)$
b_3 :	$Cl(x) \wedge BN(y) \rightarrow (x \neq y)$
b_4 :	$PI(x,y,z) \wedge \neg BN(x) \rightarrow Ind(x)$
b_5 :	$PI(x,y,z) \wedge \neg BN(y) \wedge \neg Lit(y) \rightarrow Ind(y)$
b_6 :	$PI(x,y,z) \wedge Dom(z,w) \wedge \rightarrow CI(x,w) \vee BN(x)$
b_7 :	$PI(x,y,z) \wedge Rng(z,w) \wedge \rightarrow CI(y,w) \vee (Lit(y) \wedge (w=rdfs:Literal)) \vee BN(y)$

atualização no que se refere à garantia de integridade da base. No próximo capítulo será mostrado como este mecanismo é desenvolvido no sistema BNS.

5 O SISTEMA BNS

Para a implantação da estratégia de manutenção de integridade de uma base RDF proposta no Capítulo 4, foram desenvolvidos algoritmos para computar os efeitos colaterais baseados em restrições de esquema e da aplicação em dois passos independentes. Eles consideram os seguintes princípios: (i) a partir de um conjunto de atualizações solicitado upd , as restrições de integridade são usadas para gerar uma nova lista de atualizações que devem ser aplicadas para garantir que elas sejam satisfeitas; (ii) o caracter especial “_” é usado durante o processo de inferência que computa efeitos colaterais para interromper a geração de atualizações adicionais; (iii) efeitos colaterais a partir das restrições da aplicação são computadas sem consultar a base de dados; e (iv) efeitos colaterais a partir das restrições de esquema são computados em um passo subsequente e podem exigir consulta à base de dados; (v) para que o processo possa ser executado em um único passo, ou seja, para que não seja necessário o retorno à validação das restrições de aplicação (depois do processamento das restrições do esquema RDF), foi introduzida uma etapa de tradução, para que algumas restrições de esquema sejam tratadas como restrições da aplicação.

5.1 VISÃO GERAL DO PROCESSO

O processo de atualização é dividido em dois passos principais, como ilustrado na Figura 5.1. Primeiramente, o conjunto de atualização upd é a entrada para o Algoritmo AlgAplic com objetivo de verificar as restrições de aplicação e a geração de seus efeitos colaterais. O conjunto U , gerado nesta etapa, serve de entrada para o Algoritmo AlgRDF, no qual as restrições do esquema RDF são verificadas e seus efeitos colaterais são gerados. Finalmente o conjunto U_S resultante é aplicado na base de dados D e uma nova base D' é gerada.

A meta portanto é validar as atualizações em uma abordagem feita em duas etapas, onde cada classe de restrições é tratada separadamente. No entanto, algumas restrições do esquema RDF podem exigir como efeito colateral a aplicação de restrições de aplicação. Claramente, esta situação sugere a introdução de uma iteração entre os Algoritmos AlgAplic e AlgRDF.

Para evitar esta iteração é introduzida uma etapa de Tradução (detalhada na Seção 5.2), na qual algumas restrições de esquema RDF são traduzidas pelo Algoritmo AlgTrad para o formato de restrições da aplicação (e constituem o conjunto \mathcal{M}). Estas restrições traduzidas são adicionadas ao conjunto \mathcal{C} de restrições da aplicação, para serem processadas pelo Algoritmo AlgAplic. Consequentemente, o Algoritmo AlgRDF considera somente as restrições não traduzidas (\mathcal{A}^{NT} na Figura 5.1). Esta abordagem confere melhor desempenho ao processo. A Seção 5.2 mostra que a abordagem é correta, produzindo o mesmo resultado que um processo iterativo envolvendo os Algoritmos AlgAplic e AlgRDF.

5.2 ETAPA DE TRADUÇÃO

Definição 5.1 (Tradução) Considere \mathcal{A} como sendo o conjunto de restrições de esquema RDF descrito nas Tabelas 4.3 e 4.4. Considere $\mathcal{A} = \mathcal{A}^T \cup \mathcal{A}^{NT}$ onde \mathcal{A}^T é o subconjunto de restrições a serem traduzidas em restrições da aplicação e \mathcal{A}^{NT} é o subconjunto de restrições não traduzidas. Define-se $\mathcal{A}^T = \{c \in \mathcal{A} \mid \text{body}(c) \text{ e } \text{head}(c) \text{ contém um átomo}$

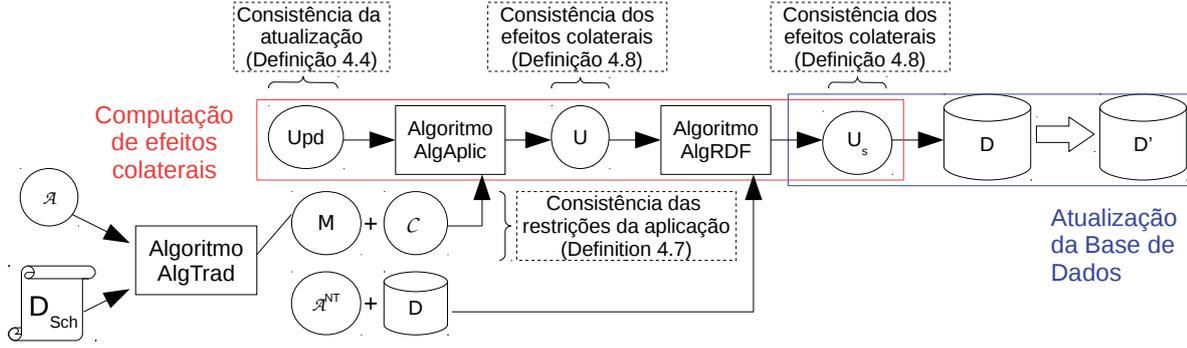


Figura 5.1: Cálculo de efeitos colaterais: requisitos de consistência em cada etapa.

com predicado CI ou PI } e $\mathcal{A}^{NT} = \mathcal{A} - \mathcal{A}^T$. A tradução é feita de acordo com o Algoritmo AlgTrad.

Algoritmo AlgTrad Etapa de Tradução

Input: O conjunto de restrições de esquema RDF (\mathcal{A}) e esquema de uma base de dados D (D_{Sch})

Output: Conjunto \mathcal{M} (restrições traduzidas)

- 1: $\mathcal{M} = \emptyset$;
 - 2: **for all** restrição $c \in \mathcal{A}$ cujo formato é $l_1 \wedge P_{cond} \rightarrow l_2 \wedge P_{alt}$, onde P_{alt} pode ser vazio e l_1 e l_2 são átomos da forma $CI(x, y)$ ou $PI(x, y, z)$ **do**
 - 3: **for all** $n_1 \in D_{Sch}$ que não contenha `rdfs:Literal` como termo e existe um homomorfismo h tal que $h(P_{cond}) = n_1$ **do**
 - 4: $\mathcal{M} = \mathcal{M} \cup \{h(l_1) \rightarrow h(l_2)\}$
 - 5: **end for**
 - 6: **end for**
 - 7: **return** \mathcal{M}
-

A partir da Definição 5.1, para cada restrição $r \in \mathcal{A}^T$ e um fato n_1 contendo um predicado de esquema existe uma função de mapeamento para traduzir r e n_1 em uma restrição da aplicação m_1 . Seja $\mu_{D_{Sch}}(r)$ o conjunto de regras geradas para uma regra r e esquema D_{Sch} . Ou seja, $\mu_{D_{Sch}}(r) = \{h(l_1) \rightarrow h(l_2) | r \text{ é da forma } l_1 \wedge P_{cond} \rightarrow l_2 \wedge P_{alt} \text{ e } h(P_{cond}) \in D_{Sch}\}$. É utilizado $\mu(r)$ ao invés de $\mu_{D_{Sch}}(r)$ quando não houver ambiguidade sobre o esquema utilizado.

Como exemplo, suponha que na base de dados D exista o fato $Csub(\text{Professor}, \text{Funcionário})$. Quando uma atualização upd insere na base $CI(\text{Joao}, \text{Professor})$, em virtude das restrições r_5 e r_6 do Exemplo 4.1, o Algoritmo AlgAplic gera como efeito colateral a remoção $\neg CI(\text{Joao}, \text{Estudante})$ e a inserção $PI(\text{Joao}, _, \text{ministra})$. No entanto, na etapa seguinte, quando o Algoritmo AlgRDF avaliar a regra m_9 da Tabela 4.3, será gerado como efeito colateral $CI(\text{Joao}, \text{Funcionario})$ e neste caso seria necessária uma nova iteração com o Algoritmo AlgAplic para verificar se houve violação de alguma regra da aplicação.

Para evitar isso, o Algoritmo AlgTrad identifica, na Linha 2, que a regra m_9 tem o formato adequado para ser traduzida, onde $l_1 = CI(x, y)$, $l_2 = CI(x, z)$ e $P_{cond} = Csub(y, z)$. Na Linha 3 é feita a busca por subclasses, ou seja fatos que casam com $Csub(y, z)$ e, por fim, na Linha 4 a regra traduzida é inserida no conjunto \mathcal{M} . Para este exemplo, será gerada a restrição $CI(X, \text{Professor}) \rightarrow CI(X, \text{Funcionario})$ e esta regra será avaliada junto com as demais regras da aplicação eliminando a iteração entre os Algoritmos AlgAplic e AlgRDF.

Além da regra m_9 as demais regras das Tabelas 4.3 e 4.4 para as quais o Algoritmo AlgTrad se aplica são: m_{10} , b_6 e b_7 . No caso da regra m_{10} , se houver na base D_{Sch} o fato $P_{sub}(Eprofessor, Efuncionario)$ então o Algoritmo AlgTrad gera a restrição $PI(X, Y, Eprofessor) \rightarrow PI(X, Y, Efuncionario)$.

No caso da regra b_6 , se na base D_{Sch} existir um fato $Dom(Eprofessor, Professor)$ então o Algoritmo AlgTrad gera a restrição $PI(X, Y, Eprofessor) \rightarrow CI(X, Professor)$ quando X não for um nodo branco.

Por fim, para a regra b_7 , se na base D_{Sch} existir um fato $Rng(Eprofessor, Universidade)$ então o Algoritmo AlgTrad gera a restrição $PI(X, Y, Eprofessor) \rightarrow CI(Y, Universidade)$ quando Y não for um nodo branco ou um literal.

Proposição 5.1 *Considere que r seja uma restrição em \mathcal{A}^T com o formato $l_1 \wedge P_{cond} \rightarrow l_2 \wedge P_{alt}$ e D_{Sch} é um esquema RDF. Considere que $\mu_{D_{Sch}}(r)$ seja sua tradução de acordo com o Algoritmo AlgTrad com um esquema D_{sch} e D uma instância da base de dados sobre D_{sch} sem nodos brancos. Então, D satisfaz $\mu(r)$ se e somente se D satisfaz r .*

Prova:

Primeiramente, considere que existe uma restrição $c \in \mathcal{A}^T$ tal que $D \models c$ mas $D \not\models \mu(c)$. Então existe uma regra m em $\mu(c)$, $m = f_1 \rightarrow f_2$ tal que $D \not\models m$. Ou seja, existe um homomorfismo h_1 tal que $h_1(f_1)$ é um fato que dispara m exigindo que $h_1(f_2)$ esteja na base de dados, mas $h_1(f_2) \notin D$. De acordo com o Algoritmo AlgTrad, a restrição c tem o formato $l_1, P_{cond} \rightarrow l_2 \wedge P_{alt}$, $f_1 = h(l_1)$ e $f_2 = h(l_2)$.

Como $D \models c$ as seguintes situações são possíveis:

(A) A regra c não é disparada porque não há um homomorfismo h tal que $h(P_{cond})$ seja um fato em D_{Sch} . Mas neste caso, o Algoritmo AlgTrad não produz a regra m em $\mu(c)$. Uma contradição com a hipótese inicial que m pertence a $\mu(c)$.

(B) $h(P_{cond})$ existe em D_{Sch} e existe um homomorfismo h_1 tal que $h_1(h(l_1)) \in D$ (uma vez que $h(l_1) = f_1$). Com a aplicação do operador T (Definição 4.5) a restrição c é disparada produzindo $h_1(f_2) \in D$ (uma vez que $h(l_2) = f_2$). Uma contradição com a afirmação inicial que $h_1(f_2) \notin D$. Existem dois casos especiais quando P_{alt} não é vazio. A primeira é quanto P_{alt} envolve $BN(x)$. Como a suposição é de que não existem nodos brancos na base D , esta regra não se aplica. A outra é quando P_{alt} envolve literais. No entanto, neste caso o Algoritmo AlgTrad não gera a regra em $\mu(r)$. Portanto, $D \models m$.

No sentido inverso, considere que existe uma restrição $c \in \mathcal{A}^T$ tal que $D \models \mu(c)$ mas $D \not\models c$. De acordo com o Algoritmo AlgTrad, as restrições em $\mu(c)$ tem o formato $f_1 \rightarrow f_2$. Considere que $D \models \mu(c)$ então existe um homomorfismo h tal que $h(P_{cond})$ existe em D_{Sch} (caso contrário, $\mu(c)$ não seria gerado). Portanto, dois casos são possíveis:

(A) Se não há um homomorfismo h_1 tal que $h_1(f_1) \in D$ então $\mu(c)$ é satisfeito trivialmente. Como $f_1 = h(l_1)$, a restrição c não é disparada da mesma forma e portanto é satisfeito trivialmente. Uma contradição com a hipótese inicial.

(B) Para todo homomorfismo h_1 tal que $h_1(f_1) \in D$, tem-se $h_1(f_2) \in D$. Em outras palavras, se $h_1(h(l_1)) \in D$, então $h_1(h(l_2)) \in D$. Uma contradição com a hipótese que $D \not\models c$. \square

5.3 ALGORITMO - RESTRIÇÕES DA APLICAÇÃO

Executada a etapa de Tradução, o Algoritmo AlgAplic é invocado para computar os efeitos colaterais impostos pelas restrições da aplicação. Neste algoritmo, a Linha 1 aplica o operador T sobre um conjunto de atualizações upd e obtém um conjunto de átomos

positivos e negativos que são os efeitos colaterais impostos por \mathcal{C} movendo da esquerda para a direita com as inserções e remoções presentes em upd . Uma vez que as requisições de atualização sejam atendidas, a consistência é verificada na Linha 2) de acordo com a Definição 4.4. Átomos positivos e negativos são verificados separadamente. Quando regras do Tipo 2 (Definição 4.2) são usadas por T , o caracter especial “_” é introduzido no átomo e portanto o resultado em $T^*(upd)$ é um conjunto de átomos e não necessariamente fatos.

Exemplo 5.1 *Considere um conjunto de atualizações $upd = \{PI(John, projDB, coordena)\}$ no Exemplo 4.3 e o conjunto de atualizações computado $T(upd)^+$. Esta inserção requer que John seja um membro de $projDB$ de acordo com r_3 ($PI(John, projDB, membro)$). Além disso, John deve ser um pesquisador pela regra r_1 ($CI(John, Pesquisador)$), e portanto um membro de algum projeto, conforme r_2 ($PI(John, _, membro)$). A regra r_4 também exige que um pesquisador seja um professor ($CI(John, Professor)$), e que um professor deve ministrar no mínimo uma disciplina, pela regra r_6 ($PI(John, _, ministra)$). A regra r_5 define que um professor não pode ser um aluno e portanto $\neg CI(John, Aluno)$ deve ser removido da base de dados caso este fato exista. Com a limpeza deste conjunto, remove-se $PI(John, _, membro)$, já que ele é subordinado a $PI(John, projDB, membro)$.*

Algoritmo AlgAplic Efeitos Colaterais devido a \mathcal{C}

Input: Um conjunto de fatos (positivos ou negativos) upd e um conjunto consistente de restrições \mathcal{C}

Output: O conjunto U de efeitos colaterais de upd relativos a \mathcal{C} .

```

1: Compute  $T := clean(T^*(upd))$ , onde  $T = T^*(upd)^+ \cup T^*(upd)^-$ 
2: if  $T$  é consistente then
3:    $Inv := \emptyset$ ;  $U := \emptyset$ ;
4:   for all  $f \in \ominus(T^*(upd)^-)$  or  $f \in T^*(upd)^+$  do
5:      $Inv := Inv \cup \vartheta^*(\neg f)$ 
6:   end for
7:   for all  $f \in \bullet(T^*(upd)^-)$  do
8:      $Inv := Inv \cup \vartheta_{|ty1,3}^*(\neg f)$ 
9:      $\ominus(U^+) := \eta(\neg f)$ 
10:  end for
11:   $U^+ = U^+ \cup T^*(upd)^+$ 
12:   $U^- = T^*(upd)^- \cup \neg.Inv$ 
13:  return  $U$ 
14: else
15:   Erro de Exceção: atualizações solicitadas não são consistentes
16: end if

```

Cada átomo positivo obtido na Linha 1 representa uma inserção ou remoção requerida. O operador ϑ é usado para encontrar átomos que geram $\neg f$ (o inverso de f). Considere as restrições do Exemplo 4.1 e a atualização $f_1 : \neg PI(John, projDB, membro)$. O resultado do operador inverso, $\vartheta(\neg f_1)^+$ contém $f_2 : PI(John, projDB, coordena)$, pelo movimento da direita para a esquerda da regra r_3 . Observe que f_1 está em $T^*(upd)^-$ e no algoritmo, f_2 é inserido em Inv na Linha 8. Na sequência, na Linha 12, $\neg f_2$ é inserido no conjunto de atualizações resultante U^- . O mesmo tratamento é dado para átomos positivos no conjunto de atualizações ($T^*(upd)^+$), e átomos negativos com nodos brancos ($\ominus(T^*(upd)^-)$) (Linhas 4 e 5).

Continuando o exemplo, $f_3 : CI(John, Pesquisador)$ também está em $\vartheta(\neg f_1)$ pela regra r_2 . No entanto, a regra r_2 é do Tipo 2. Neste caso, o algoritmo não inclui f_3 em Inv (uma vez que o operador ϑ é restrito a regras do Tipo 1 e 3), mas aplica o operador η (Linha 8) em f_1 e inclui $f_4 : PI(John, _, membro)$ em Inv . O conjunto de atualizações computado pelo Algoritmo AlgAplic satisfaz algumas propriedades, definidas pela seguinte proposição.

Proposição 5.2 *Considere que upd seja um conjunto de atualizações consistente (Definição 4.4) e \mathcal{C} seja um conjunto consistente de restrições (Definição 4.7). O conjunto de atualizações U obtido pelo Algoritmo AlgAplic satisfaz as seguintes propriedades:*

1. $CleanedUpd \subseteq upd$ **E** $CleanedUpd \subseteq U$ onde
 $CleanedUpd = \{op_1 \in upd \mid \nexists op_2 \text{ tal que } op_1 \preceq op_2 \text{ e } op_2 \in T^*(upd)\};$
2. U é consistente em relação à Definição 4.8;
3. U é válido em relação a \mathcal{C} ;
4. O conjunto de efeitos colaterais computado é mínimo, ou seja, para cada átomo $l \in (U \setminus CleanedUpd)$ o conjunto $D \uplus (U \setminus \{l\})$ não satisfaz \mathcal{C} ($D \uplus I$ é a aplicação da atualização I na base D conforme definido na Seção 4.2).

Prova:

1. $CleanedUpd \subseteq upd$ **E** $CleanedUpd \subseteq U$

Pela definição de $CleanedUpd$, sabe-se que $CleanedUpd \subseteq upd$.

Seja f um fato positivo em $CleanedUpd$. De acordo com a Definição 4.5, $f \in T^*(upd)$. Para cada $l \in T^*(upd)$ tal que $\exists h$ (um homomorfismo) e $f = h(l)$, ou $l = f$ ou l tem o caracter “_”, e de acordo com a Definição 4.3, $l \preceq f$. Portanto, l é removido pela função $Clean$. Pode-se concluir que para todos os fatos positivos f em upd , tem-se que $f \in Clean(T^*(upd))$ e $f \in U$.

Agora considere um fato negativo $\neg f$ em $CleanedUpd$. De acordo com a Definição 4.5, $\neg f \in T^*(upd)$. Mas se há um átomo negativo instanciado $\neg f_1 \in T^*(upd)$ tal que $\neg f = h(\neg f_1)$, de acordo com a Definição 4.3, $\neg f \preceq \neg f_1$ e $\neg f$ é removido pela função $Clean$. Uma vez que $\neg f \in CleanedUpd$, não há um átomo tal que $\neg f_1 \in T^*(upd)$. Portanto todos os $\neg f \in CleanedUpd$ também estão em U .

2. U é consistente em relação à Definição 4.8.

A segunda propriedade é provada por contradição considerando que há literais positivos l e l' tal que $l \in U^+$, $\neg.l' \in U^-$ e $l = h(l')$. Analisando as Linhas 11 e 12 do Algoritmo AlgAplic, observa-se que há 4 situações a considerar:

- (1) $l \in T^*(upd)^+$ e $\neg.l' \in T^*(upd)^-$ o que é impossível devido ao teste na Linha 2.
- (2) $l \in T^*(upd)^+$ e $\neg.l' \in \neg.Inv$. Primeiramente, considere que $\neg.l' \in \neg.Inv$ porque há um literal f tal que $l' \in \vartheta^*(\neg.f)$. Neste caso, existe um fato $f_0 \in upd$ tal que $f \in T^*(f_0)$. Há também um fato $l_0 \in upd$ tal que $l \in T^*(l_0)$. Como $\neg.f \in T^*(l')$ e $l = h(l')$, sabe-se que $\neg.f \in T^*(l_0)$. Como f_0 e l_0 estão ambos em upd , há uma contradição como o fato que upd é consistente (teste na Linha 2). Agora, considere que $\neg.l' \in \neg.Inv$ porque há um literal negativo $l_1 \in \ominus T^*(upd)^-$ tal que $l' \in \vartheta^*(\neg.l_1)$. Portanto existe $f_1 \in upd$ para o qual $l_1 \in \ominus T^*(f_1)^-$. Como há também um fato $l_0 \in upd$ para o qual $l \in T^*(l_0)$ e $l = h(l')$, o conjunto upd não

- é consistente (uma vez que a partir de l_0 e f_1 são gerados literais contraditórios $\neg.l_1$ e l_1 , respectivamente). Esta é uma contradição em relação à suposição original.
- (3) $l \in \ominus(U^+)$ e $\neg.l' \in T^*(upd)^-$. Neste caso, há um literal negativo $\neg.f_2 \in \bullet(T^*(upd)^-)$ e de acordo com a Definição 4.5 $f_2 = h_1(l)$ onde h_1 é um homomorfismo que mapeia o nulo de l (na posição existencial) em uma constante que aparece em f_2 na mesma posição (Linhas 7 e 9). Neste caso $\neg.l'$ e f_2 estão em U , mas sabe-se que $f_2 = h_1(h(l'))$. Portanto, $T^*(upd)$ deriva $\neg.f_2$ e $\neg.l'$ e l' está subordinado a f_2 , uma contradição em relação à Definição 4.8.
- (4) $l \in \ominus(U^+)$ e $\neg.l' \in \neg.Inv$. A prova combina as condições verificadas nos itens (2) e (3) acima.

3. U é válido em relação a \mathcal{C} .

Para provar a terceira propriedade assuma que existe uma restrição em \mathcal{C} que não é satisfeita por U . Ou seja, existe uma restrição $c : \alpha \rightarrow \beta$, um fato $f \in U^+$ e um homomorfismo h e $h'(h \subseteq h')$ tal que $f = h(\alpha)$ and $h'(\beta) \notin U$. Neste caso, f não pode estar em $T^*(upd)$ (ou todos os átomos instanciados em $T^*(upd)$ estão em U ou existe um fato removido de U , mas submisso a um outro fato $\in U$). Se f está em Inv então existe um átomo negativo $\neg.f_1$ em U capaz de produzir f a partir de ϑ em um ou mais passos. E portanto existe uma restrição c capaz de produzir um fato $h'(\beta) \in U$ iniciado com $f = h(\alpha)$. Uma contradição com relação a hipótese inicial. Se f é um átomo resultante do uso do operador η (Linha 9), isto é uma contradição já que, neste caso, f não tem correspondência com o corpo de nenhuma restrição em \mathcal{C} .

4. O conjunto de efeitos colaterais computado é mínimo, ou seja, para cada átomo $l \in (U \setminus CleanedUpd)$ o conjunto $D \uplus (U \setminus \{l\})$ não satisfaz \mathcal{C} .

Se $l \in (U \setminus CleanedUpd)$, ele foi gerado como efeito colateral de uma restrição de aplicação.

Há 3 possibilidades para considerar:

(1) l é um efeito colateral em $Clean(T^*(upd))$. Neste caso existem duas possibilidades:

(1a) l foi gerada pela regra do tipo $L_1(X) \rightarrow L_2(X, Y)$. Neste caso l tem o caracter “_”. Se $l \in U \setminus \{CleanedUpd\}$ então não há um fato $f \in U$ tal que $l \preceq f$ e não há l_1 tal que l_1 seja isomórfico a l . Portanto, se l for eliminado de U , $U \setminus \{l\} \not\models \mathcal{C}$

(1b) l foi gerado por uma regra do tipo $(L_1(X) \rightarrow L_2(X))$.

Neste caso l não possui “_” e não há l_1 tal que l_1 seja isomórfico a l . Portanto $U \setminus \{l\} \not\models \mathcal{C}$.

(2) l foi gerado por $\vartheta^*(\neg.f)$ (Linha 5 ou 8);

Neste caso, $l = \neg.l_1$ é um átomo que deseja-se remover. Portanto há um fato f gerado por T usando apenas regras do tipo 1 ou 3 tal que $f \in U$. Na Linha 5, o Algoritmo AlgAplic faz a remoção de fatos que contradizem f . Um destes átomos é $l_1 \in \vartheta^k(\neg.f)$ para um passo $k > 0$ da computação de ponto fixo. Portanto, há regras $\alpha_1 \rightarrow \alpha_2 \rightarrow \dots \alpha_n \rightarrow \beta$ e existe um homomorfismo h tal que $\neg.f = h(\beta)$ e $\exists h' (h \subseteq h')$ tal que, para cada $1 \leq i \leq n$, $h'(\alpha_i)$ é uma instanciação para uma restrição $c \in \mathcal{C}$. Assuma $l_1 = h'(\alpha_k)$ para um $k \in [1, n]$.

Lembrando que $U \setminus \{l\} = U \setminus \{\neg l_1\}$, o que significa que em U não existe uma atualização que exija a remoção de l_1 . Considere $D \uplus (U \setminus \{\neg l_1\})$. O banco de dados resultante pode ter um átomo instanciado $f_1 = l_1 = h'(\alpha_k)$ para uma restrição $\alpha_k \rightarrow \alpha_{k+1}$ e portanto $\neg f$ será gerado. Como $f \in U$, $D \uplus (U \setminus \{\neg l_1\})$ será inconsistente.

(3) l foi gerado por $\eta(\neg f)$ (Linha 9)

Se l foi gerado por $\eta(\neg f)$, sabe-se que existe uma restrição $c : \alpha \rightarrow \beta$ do tipo 2 e um homomorfismo $h(\alpha) \in U$ e $h'(\beta) = f$.

Além disso, nesta situação $\neg f \in U$ (já que a situação acontece quando a remoção de f é exigida) e $f \notin U$ (devido a checagem de consistência na Linha 2). Operador $\eta(\neg f)$ garante a presença de um átomo l (que contém “_”) tal que $f = h_1(l)$. Sem o átomo instanciado l , a base de dados atualizada $D \uplus (U \setminus \{l\})$ pode não satisfazer \mathcal{C} (e U não satisfaz \mathcal{C}).

Note que se há $l_3 \in D$ tal que $l_3 = h_1(l)$, l não é realmente necessário mas será instanciado por l_3 . Se tal l_3 não existe, l é necessário para manter a consistência.

□

5.4 ALGORITMO - RESTRIÇÕES RDF

Na última etapa do processo, o Algoritmo AlgRDF computa os efeitos colaterais baseados em restrições do esquema RDF nas Tabelas 4.3 e 4.4, excetuando as restrições traduzidas durante a etapa de Tradução e já verificadas pelo Algoritmo AlgAplic.

O Algoritmo AlgRDF computa U_S , uma extensão de U que inclui: (1) a interpretação para cada operação de atualização envolvendo instâncias, conforme a Tabela 4.2 e (2) efeitos colaterais obtidos pela aplicação de \mathcal{A}^{NT} em U . Isso é realizado pela função *ResultTrigRule*. A função considera as regras \mathcal{A}^{NT} das Tabelas 4.3 e 4.4. De forma similar ao Algoritmo AlgAplic, as inserções (átomos positivos) ativam regras da esquerda para a direita e a instanciação da cabeça da regra é adicionada a U , enquanto remoções (átomos negativos) ativam regras da direita para a esquerda e o inverso da instanciação do corpo da regra é inserido em U .

A Linha 3 rejeita conjuntos de atualização que contêm alteração de esquema ou que são inconsistentes em relação à Definição 4.8. Como exemplo, considere a inserção de $CI(CNPq, RInst)$. A função *ResultTrigRule* aplica a regra m_7 e gera como efeito colateral em U a inserção da classe $RInst$ ($Cl(RInst)$). Caso este fato não exista previamente em D , esta operação determina uma alteração no esquema da base. Como este trabalho não considera alterações de esquema, o conjunto de atualização é rejeitado pelo Algoritmo AlgRDF.

Na Linha 8, o conjunto U é completado da seguinte forma pela função *ResultTrigRule*. Considere \mathcal{C} do Exemplo 4.1 e a instância da base de dados $\{CI(Bob, Professor), CI(BD, Curso), PI(Bob, BD, ministra)\}$. A atualização $upd = \{\neg Ind(BD)\}$ produzirá a $\neg CI(BD, Curso)$ pelo Algoritmo AlgRDF, considerando a aplicação da regra m_6 . Além disso, a regra b_5 produz $\neg PI(Bob, BD, ministra)$. Tal remoção viola r_6 (o que é identificado na Linha 10), que determina que todo professor deve ministrar pelo menos uma disciplina. Portanto, o algoritmo adiciona $PI(Bob, _, ministra)$ a U . Note que esta operação adicionará um nulo se BD era o único curso ministrado por Bob , conforme a semântica de $\llbracket PI(Bob, _, ministra) \rrbracket_D$ definida na Tabela 4.2.

Algoritmo AlgRDF Efeitos colaterais devido a \mathcal{A}

Input: Um conjunto de atualizações U , uma instância da base de dados D , restrições Σ ($\mathcal{A} \cup \mathcal{C}$)

Output: O conjunto U_S de efeitos colaterais de U com relação a Σ .

```

1:  $U_S := \emptyset$ ;
2: repeat
3:   if  $ChangeSchema(U, D)$  ou  $U$  não é consistente em relação à Definição 4.8 then
4:     Erro;
5:   end if
6:    $U_0 := U$ ;
7:    $U_S := U_S \cup \{[op]_{D \uplus U_S} \mid op \in U \wedge op \text{ tem um predicado em INSTPRED}\}$ ;
8:    $U := U \cup \{l \mid l = ResultTrigRule(r, op, D \uplus U_S) \text{ tal que } r \in \mathcal{A}^{NT} \wedge op \in U_0\}$ ;
9:   for all remoções da forma  $\neg Ind(a)$ ,  $\neg PI(a, \_, p)$  ou  $\neg PI(\_, a, p)$  em  $U_0$  do
10:     $U := U \cup \{l \mid l \text{ é da forma } PI(\_, b, p) \text{ (ou } PI(b, \_, p)), \text{ tal que existe } PI(a, b, p) \text{ (} PI(b, a, p), \text{ respect.) em } (D \uplus U_S) \text{ cuja remoção viola } r \in \mathcal{C}\}$ ;
11:   end for
12: until  $U = U_0$ ;
13: return  $U_S$ ;

```

Nas Linhas 9 e 10 é feito um tratamento especial para a remoção de indivíduos e da remoção de relacionamentos com $_$. Isto se deve às restrições m_6 (Tabela 4.3), b_4 e b_5 (Tabela 4.4). Estas restrições não atendem o formato definido no Algoritmo AlgTrad para serem traduzidas. No entanto, a ativação delas gera efeitos colaterais que podem violar restrições da aplicação, o que exigiria a reexecução do Algoritmo AlgAplic. Assim, este tratamento faz a identificação dos relacionamentos existentes na base de dados que envolvam o indivíduo a que está sendo removido, e se existem restrições de aplicação em \mathcal{C} que obriguem a existência de tais relacionamentos para o indivíduo b que permanece na base de dados.

Além disso, como a semântica de $PI(_, b, p)$ (ou $PI(a, _, p)$) definida na Tabela 4.2 determina a remoção de todos os relacionamentos de a através do predicado p . Existe o mesmo risco de remoção de um relacionamento obrigatório conforme explicado no parágrafo anterior. Portanto neste caso também é incluído um relacionamento com $_$.

Proposição 5.3 *Considere que $\Delta = (D, D_{Sch}, \Sigma)$ seja uma base de dados consistente em relação a $\Sigma = (\mathcal{C}, \mathcal{A})$. Dado um conjunto de atualizações upd , considere que U_S seja o conjunto de efeitos colaterais computados de acordo com os Algoritmos AlgAplic e AlgRDF. Considere que U_S^{Sch} seja o subconjunto de U_S com fatos em SCHPRED. Se U_S é consistente (Definição 4.8) e $U_S^{Sch} \subseteq D_{Sch}$ então o resultado de $D \uplus U_S$ é uma nova instância da base de dados D' que satisfaz as seguintes propriedades: (i) $upd^+ \subseteq D'$; (ii) $upd^- \cap D' = \emptyset$ e (ii) D' satisfaz \mathcal{C} e \mathcal{A} . \square*

Prova:

1. $upd^+ \subseteq D'$

De acordo com a Proposição 5.2, todos os fatos $f \in upd^+$ estão em U (conjunto de entrada para o Algoritmo AlgRDF), uma vez que estes fatos não tem nulos. Além disso, o algoritmo inclui em U_S a semântica destas operações, que, de acordo com a Tabela 4.2, contem o próprio fato f . Portanto, um fato $f \in upd^+$ não está

em U_S apenas se o Algoritmo AlgRDF gera $\neg f$. Neste caso, o algoritmo para na Linha 3 devido à checagem de consistência. Portanto, $upd^+ \subseteq U_S$. Dada a semântica do operador \uplus estes fatos positivos já estão em D , ou inseridos em D' . Portanto, $upd^+ \subseteq D'$.

2. $upd^- \cap D' = \emptyset$

De acordo com a proposição 5.2, cada fato negativo $\neg f$ in upd^- ou está em U ou ele é submisso a um outro fato $\neg f_1$ in U . Em qualquer dos casos, na Linha 7, o algoritmo segue a computação da semântica de atualização indicada na Tabela 4.2, para incluir a interpretação do fato em U_S . Se $\neg f \in U$ a interpretação é o próprio fato $\neg f$. Se $\neg f_1$ está em U então (1) $\neg f$ está na interpretação $[[\neg f_1]]_{D \uplus U_S}$ e $\neg f$ é inserido em U_S , ou (2) $f \notin D$, e U_S não contém f . No primeiro caso, depois de aplicar o conjunto de atualização U_S em D , f será removido. No segundo caso, f não fazia parte da base de dados original e, portanto, $upd^- \cap D' = \emptyset$.

3. D' satisfaz \mathcal{C} .

O conjunto U , que satisfaz \mathcal{C} , conforme provado na Proposição 5.2, é a entrada para o Algoritmo AlgRDF. O papel do Algoritmo AlgRDF é computar o conjunto U_S que contém todas as atualizações a serem aplicadas à base de dados como consequência de uma atualização upd solicitada pelo usuário. Sua construção se inicia com U (computado no Algoritmo AlgAplic). Então, conforme é processado o Algoritmo AlgRDF, outros efeitos colaterais podem ser gerados, causados inicialmente pelas atualizações em U . Estes novos efeitos colaterais são resultantes da semântica de atualizações definidas na Tabela 4.2 e a partir da aplicação das regras em \mathcal{A}^{NT} .

Para provar que $D' \models \mathcal{C}$ deve-se provar que aplicar todas as atualizações de U_S em D resulta em uma base de dados D' consistente.

(I) Inicialmente serão consideradas as *inserções*. Quando a inserção de um átomo A é exigida em U_S , tem-se uma das seguintes situações:

(A) *O átomo A está no conjunto de entrada U .*

Neste caso, a inserção do átomo A coincide com uma requisição em upd ou um efeito colateral já tratado pelo Algoritmo AlgAplic. Em ambos os casos,

(i) se a inserção ativa regras em \mathcal{C} , seus efeitos colaterais já foram avaliados pelo Algoritmo AlgAplic, e estão em U , que satisfaz \mathcal{C} (Proposição 5.2). Portanto U contém todas as atualizações necessárias para manter a consistência devido a inserção de A ;

(ii) se há restrições impostas ao átomo A que não possam coexistir na base de dados D com um átomo $B \in D$ (ou seja, $B \rightarrow \neg A$)¹, o Algoritmo AlgAplic garante que a delegação de cada átomo B e todos os seus efeitos colaterais são requeridos em U . No entanto, quando um átomo B (sendo removido) tem a forma $CI(_, C)$ trata-se de uma situação especial que é considerada no item 3(II)Bb mais adiante.

¹Em geral, pode haver uma sequência de restrições $r_1 \dots r_n$ tal que há um homomorfismo h tal que $h(\text{corpo}(r_1))=B$, $h(\text{corpo}(r_n))=\neg A$ e para todo $1 \leq i \leq n$, $h(\text{head}(r_{i-1})) = \text{body}(r_i)$. No entanto, o resultado é o mesmo: a presença de B impõe a ausência de A em uma instância consistente.

(B) O átomo A não está no conjunto de entrada U , mas ele é gerado pelo Algoritmo AlgRDF na linha 7, pela aplicação da semântica de uma outra atualização já em U_S , que está em construção.

Primeiramente, se consideradas apenas as inserções, o átomo A que respeita as condições acima não pode ser um átomo instanciado (sem nulos).

- A aplicação da semântica das inserções de um átomo da forma $CI(a, C)$ retorna a inserção do próprio átomo e corresponde à situação descrita no item (A) acima.

- A aplicação da semântica de inserção de um átomo A da forma $PI(a, b, p)$ retorna:

- (i) por um lado, a inserção do próprio átomo (correspondendo à situação descrita no item (A) acima) e,

- (ii) por outro lado, a remoção de átomos $PI(a, N_1, p)$ ou $PI(N_1, b, p)$ juntamente com $BN(N_1)$ quando eles existem na instância da base de dados D . Em outras palavras, a inserção de um átomo instanciado A exige a remoção de todos os átomos α que contenha nulos para o qual há um homomorfismo h tal que $h(\alpha) = A$. Além disso, a semântica da atualização dá prioridade para átomos instanciados, substituindo nulos por constantes quando possível. Como a instância original da base de dados D é consistente e o conjunto de restrições não é ativado por átomos que tenham nulos (nenhum corpo de regra é ativado por um átomo que tenha nulo, conforme a Definição 4.2), substituir um átomo que tem um nulo por seu equivalente não nulo não afeta a consistência da base de dados. Esta operação corresponde ao uso do algoritmo *core* [FKP05] em um caso bem simples, já que não há propagação de nulos.

A partir da discussão acima, sabe-se que quando considerada a inserção de um átomo A , se $A \notin U$ mas A é computada na linha 7, pela aplicação da semântica de uma outra atualização que já esteja em U_S , então este átomo A tem a forma $PI(_, b, p)$ ou $PI(a, _, p)$. Neste caso a aplicação da semântica na linha 7 resulta em uma das seguintes ações:

- (i) Se existe um homomorfismo h e um átomo instanciado $A' \in D$ tal que $A' = h(PI(_, b, p))$ (ou $A' = h(PI(a, _, p))$) então a inserção não é realizada, isto é, o átomo $PI(N, b, p)$ (ou $PI(a, N, p)$) não é adicionado a U_S . A base de dados atualizada D' não conterá A mas sua consistência é garantida pela presença de A' que já está na instância consistente D .

- (ii) Se não existe um átomo instanciado A' em D , o átomo $PI(N, b, p)$ (ou $PI(a, N, p)$) é adicionado a U_S juntamente com o átomo $BN(N)$. Note que, neste caso, a inserção requerida de A :

- ★ pode estar em U devido a ativação de uma restrição do Tipo 2 (em restrições do Tipo 1 e 3 não ocorre existencial na cabeça da regra) ou
- ★ pode ser um exigência decorrente da linha 10 do Algoritmo AlgRDF, tratado no item (D) abaixo.

Portanto, em ambos os casos, todos os possíveis efeitos colaterais gerados a partir da inserção de A também estão em U , como explicado no item 3(I)A acima. A inserção de $BN(N_1)$ não gera consequências para a

consistência da base de dados, uma vez que eles são independentes uns dos outros.

- (C) *O átomo A não está no conjunto de entrada U , mas ele é gerado pelo Algoritmo AlgRDF na linha 8, pelas regras ativadas em \mathcal{A}^{NT} .*

Na linha 8 as restrições do esquema RDF em \mathcal{A}^{NT} (veja Tabelas 4.3 e 4.4) são consideradas. Quando considera-se *inserções* de A , uma das seguintes situações é possível:

(i) A é uma instância de classe ou propriedade: as regras m_6 a m_8 e b_4 a b_5 podem ser ativadas. Elas podem gerar efeitos colaterais referentes à inserção de átomos do predicado *Ind* ou de SCHPRED. Inserção de indivíduos não afetam a consistência da base de dados, inserção de átomos com predicados em SCHPRED é possível apenas se estes átomos já existem na definição do esquema da base de dados. A verificação é feita na linha 3. Inserções que representam evolução de esquema não são permitidas - a atualização como um todo é rejeitada.

NOTA: É importante lembrar que as regras m_9 , m_{10} (Tabela 4.3), b_6 e b_7 (Tabela 4.4) estão em \mathcal{A}^T e portanto foram incluídas em \mathcal{C} (elas não são ativadas na linha 8).

(ii) A é um átomo de predicado de esquema. Neste caso, as regras que possivelmente podem ser ativadas podem gerar átomos em predicados de esquema ou indivíduos. Como no item (i) acima, a inserção de indivíduos não afetam a consistência da base de dados e a inserção de átomos em predicados de esquema é verificada na linha 3.

(iii) A é um indivíduo: regras m_5 ou b_2 podem ser ativadas, mas seus possíveis efeitos colaterais não afetam a consistência da base de dados.

- (D) *O átomo A não está no conjunto de entrada U , mas ele é gerado pelo Algoritmo AlgRDF na linha 10, como um efeito colateral da remoção de um indivíduo ou da remoção de instâncias de propriedades com nulos.*

Na linha 10 do Algoritmo AlgRDF dois tipos de remoção são considerados:

- A remoção de um indivíduo x ($\neg Ind(x)$) e
- Remoções da forma $PI(_, x, p)$ ou $PI(x, _, p)$

Ambos os tipos de remoção podem introduzir inconsistência na base de dados, exigindo inserções como efeitos colaterais para reestabelecer a consistência.

Mais precisamente, considere estes dois tipos de remoção:

- a) De acordo com a Tabela 4.2 a remoção de um indivíduo x ($\neg Ind(x)$) implica:

- (i) a remoção de todas as classes associadas a x ($\neg CI(x, C)$) e
- (ii) a remoção de propriedades envolvendo x ($\neg PI(x, b, p)$), para todo $PI(x, b, p) \in D$ e $\neg PI(a, x, p)$, para todo $PI(a, x, p) \in D$.

- b) Similarmente, de acordo com a Tabela 4.2, remoções da forma $PI(_, x, p)$ ou $PI(x, _, p)$ também implicam na remoção de todas as propriedades p envolvendo x . Note que uma remoção tal que $PI(_, x, p)$ (or $PI(x, _, p)$) pode ser o efeito colateral de aplicação de uma regra do Tipo 2. Na linha 7 do Algoritmo AlgRDF, a aplicação da semântica de remoção gera o conjunto de remoções de propriedades

envolvendo x ($\neg PI(a, x, p)$, para todo $PI(a, x, p) \in D$ ou $\neg PI(x, b, p)$, para todo $PI(x, b, p) \in D$).

Em ambos os casos, no entanto, a remoção de todas estas propriedades podem introduzir inconsistência uma vez que elas poderiam ser exigidas para satisfazer restrições da aplicação impostas ao indivíduo b (ou a) ligado a x por p . Para evitar inconsistência nesta situação, o Algoritmo AlgRDF, na linha 10, gera como efeito colateral a *inserção* de $PI(_, b, p)$ ou $PI(a, _, p)$ quando as seguintes condições ocorrem:

- (i) Há uma restrição $r_1 : \alpha \rightarrow \beta$ do Tipo 2;
- (ii) $PI(a, b, p)$ está sendo removida como efeito colateral de uma outra remoção e
- (iii) Este um homomorfismo h tal que $PI(a, b, p) = h(\beta)$.

De acordo com a Definição 4.2, não existe uma restrição $r_2 \in \mathcal{C}$ tal que a cabeça(r_1) e o corpo(r_2) sejam unificáveis. Portanto, a inserção de $PI(_, b, p)$ (ou $PI(a, _, p)$) não casa com nenhuma outra regra. Além disso, para a remoção de um indivíduo x , todas as propriedades e tipos de x são removidos e portanto, nenhuma restrição do Tipo 1 e 3 são violadas.

Portanto, na situação acima, a consistência da instância da base de dados é preservada.

(II) Considere agora as *remoções*. Quando a remoção de um átomo A é exigida em U_S , tem-se uma das seguintes situações:

(A) *O literal $\neg A$ está no conjunto de entrada U .*

Neste caso a remoção do átomo A coincide com a solicitação do usuário em *upd* ou com um efeito colateral já tratado pelo Algoritmo AlgAplic. A prova é similar ao item 3(I)A. Em ambos os casos,

- (i) se a remoção ativa regras em \mathcal{C} , seus efeitos colaterais já foram avaliados pelo Algoritmo AlgAplic, e estão em U que satisfaz \mathcal{C} (Proposição 5.2). Portanto U contém todas as atualizações necessárias para manter a consistência devido a remoção de A ;
- (ii) se existem restrições impondo que A deveria existir com um átomo B já em D , o Algoritmo AlgAplic garante que a remoção de B e todos seus efeitos colaterais estão em U . Além disso, quando B tem a forma $CI(_, C)$ há uma situação especial que é considerada no item 3(II)Bb abaixo.

(B) *O literal $\neg A$ não está no conjunto de entrada U , mas ele é gerado pelo Algoritmo AlgRDF na linha 7, pela aplicação da semântica de uma outra atualização já presente em U_S , que está em construção.*

- a) A aplicação da semântica de remoção de um átomo da forma $CI(a, C)$ (ou $PI(a, b, p)$) retorna a remoção do próprio átomo, ou seja, nenhum outro efeito colateral (além daqueles já computados pelo Algoritmo AlgAplic) é produzido.
- b) A aplicação da semântica de remoção de um átomo A da forma $CI(_, C)$ não causa alteração na base de dados.

Neste caso, a consistência pode ser violada se as seguintes condições ocorrerem:

- Existe uma restrição da forma $CI(x, C) \rightarrow PI(x, y, p)$;
- A instância da base de dados contém uma instância α de $PI(x, y, p)$
- A remoção de α é requerida por alguma outra restrição (por exemplo, suponha a inserção de $CI(c', C')$ que gera $\neg\alpha$ como efeito colateral)

De fato, neste caso, a consistência de U (computada pelo Algoritmo AlgAplic) é garantida pois $\neg CI(_, C)$ está em U . No entanto, como a semântica de remoção de $CI(_, C)$ não tem efeito na instância da base de dados, se, por exemplo, um fato como $CI(c_1, C)$ existe na instância da base de dados, ele não é removido. Nesta situação, a base de dados estará inconsistente. Este estado inconsistente é evitado na linha 10, pela inserção de um átomo que tenha a forma $PI(_, y, p)$, uma situação considerada no item 3(I)D.

- c) A aplicação da semântica de remoção de um átomo da forma $PI(_, b, p)$ (ou $PI(a, _, p)$) retorna a remoção de todas as propriedades p envolvendo b (ou a). Todas estas remoções são inseridas em U (linha 7) juntamente com seus efeitos colaterais. No item 3(I)Db foi provado como a consistência é preservada neste caso.
- d) A aplicação da semântica de remoção de um átomo da forma $PI(a, N1, p)$ (ou $PI(N1, b, p)$) implica na remoção do próprio átomo. Um átomo que tenha esta forma está em uma instância da base de dados como o resultado da aplicação de inserções tais como $PI(a, _, p)$ (ou $PI(_, b, p)$) conforme indicado na Tabela 4.2. O átomo armazenado tem um valor branco $N1$ e não um caracter “_”. A inserção de um átomo da forma $PI(a, b, p)$ tem como consequência a remoção de átomos da forma $PI(a, N1, p)$ (ou $PI(N1, b, p)$) que existam na base de dados, além da remoção de $BN(N1)$ que está associado a ele ². Em outros termos, um átomo instanciado substitui aquele que tenha nulos. Esta remoção não tem efeito na consistência da base de dados, ela corresponde a uma simples aplicação do algoritmo *core*[FKP05].

(C) *O literal $\neg A$ não está no conjunto de entrada U , mas ele é gerado pelo Algoritmo AlgRDF na linha 8, pela ativação de regras em \mathcal{A}^{NT} .*

Na linha 8 as restrições RDF em \mathcal{A}^{NT} são consideradas (veja Tabelas 4.3 e 4.4). No entanto, alterações no esquema não são permitidas. Portanto, apenas as regras $m6$, $b4$ e $b5$ podem ser ativadas como consequência da remoção de um indivíduo $Ind(a)$. A remoção de indivíduos é considerada na linha 10 do Algoritmo AlgRDF. A prova de consistência nesta situação é dada no item 3(I)D.

(D) *O literal $\neg A$ não está no conjunto de entrada U , mas ele é gerado pelo Algoritmo AlgRDF na linha 10, como um efeito colateral da remoção de um indivíduo ou da remoção de instâncias de propriedades com nulos.*

A prova de consistência nesta situação é dada no item 3(I)D.

²Como nulos são independentes, nenhum outro átomo envolve $N1$ e portanto a remoção de $BN(N1)$ não produz efeitos colaterais.

Assim, foi provado que o Algoritmo AlgRDF considera todas atualizações em U e gera todas as atualizações necessárias para aplicação da semântica de atualizações descrita na Tabela 4.2 ou para garantir a consistência da instância da base de dados. \square

Por fim, é preciso retornar à Etapa de Tradução e apresentar a última proposição:

Proposição 5.4 *Para toda restrição $c \in \mathcal{A}$, se a ativação de c propicia a ativação de uma restrição $c_1 \in \mathcal{C}$, então (1) $c \in \mathcal{A}^T$; ou (2) c é tratado isoladamente no Algoritmo AlgRDF, eliminando a necessidade de reexecutar o Algoritmo AlgAplic.*

Prova: O caso (2) foi demonstrado na Proposição 5.3. Assim, nesta prova será considerado o caso (1):

Suponha que existe uma restrição $c : a \rightarrow b$ em \mathcal{A} cuja ativação resulta na ativação da restrição $c_1 : a_1 \rightarrow b_1$ em \mathcal{C} . É preciso analisar duas situações:

(A) c é ativada por uma inserção. Neste caso, há um homomorfismo h tal que $b = h(a_1)$. De acordo com a Definição 4.2, se $b = h(a_1)$ então b é um átomo da forma $CI(x, y)$ ou $PI(x, y, z)$.

Conforme o Algoritmo AlgTrad, uma restrição da semântica RDF $c : l_1 \rightarrow l_2$ é traduzida para uma restrição da aplicação se l_1 e l_2 são átomos da forma $CI(x, y)$ ou $PI(x, y, z)$. A regra m_3 (Tabela 4.3) merece atenção, já que é a única restrição em \mathcal{A} onde a cabeça é da forma $CI(x, y)$ e não é uma candidata para a tradução (uma vez que o corpo de m_3 não respeita a condição imposta pelo Algoritmo AlgTrad). No entanto, mesmo que m_3 seja ativada, não há homomorfismo possível de $CI(x, rdfs : resource)$ com nenhuma outra restrição em \mathcal{C} , já que m_3 define que todos os indivíduos são instâncias de recursos, não seria útil usar $CI(x, rdfs : resource)$ em nenhuma regra da aplicação.

Para todas as outras restrições $c \in \mathcal{A}$ cujo corpo respeita a condição imposta pelo Algoritmo AlgTrad, a única razão para evitar sua tradução pelo Algoritmo AlgTrad é o fato de P_{cond} não ser satisfeita por D_{Sch} . No entanto, se P_{cond} não é satisfeita, a restrição c não é disparada, uma contradição à hipótese inicial.

(B) c é ativada por uma remoção. Neste caso, há um homomorfismo h tal que $a = h(b_1)$. De acordo com a Definição 4.2, se $a = h(b_1)$ então a é um átomo da forma $CI(x, y)$ ou $PI(x, y, z)$.

De acordo com o Algoritmo AlgTrad, uma restrição da semântica RDF $c : l_1 \rightarrow l_2$ é traduzida em uma restrição da aplicação se l_1 e l_2 são átomos da forma $CI(x, y)$ ou $PI(x, y, z)$.

As restrições m_7 e m_8 em \mathcal{A} não são traduzidas porque suas cabeças não respeitam a condição do Algoritmo AlgTrad ainda que seus corpos o façam. No entanto, tais restrições podem ser ativadas apenas por remoções no esquema – uma operação que não é permitida neste trabalho.

As restrições m_6 , b_4 e b_5 não são traduzidas (suas cabeças não respeitam a condição exigida pelo Algoritmo AlgTrad). A possível ativação destas restrições é considerada no Algoritmo AlgRDF – Linhas 9 e 10 cuja meta é evitar a ativação de $c_1 \in \mathcal{C}$ incluindo nodos brancos quando necessário.

As restrições m_9 , m_{10} , b_6 e b_7 não são traduzidas apenas se P_{cond} no Algoritmo AlgTrad não é satisfeita. Neste caso, se P_{cond} não é satisfeita, a restrição c_1 não é ativada, uma contradição com a hipótese inicial. \square

Complexidade. O Algoritmo AlgTrad tem seu tempo limitado à quantidade de regras em \mathcal{A} e para cada uma delas a busca por P_{cond} em D ($O(|\mathcal{A}| \times |D|)$), onde $|D|$ é o tamanho da base de dados. No Algoritmo AlgAplic a computação do ponto fixo de T

corresponde a uma consequência imediata do operador usado em Datalog, uma vez que apenas regras positivas podem iterar. Sabe-se que o número de iterações é limitado pelo número de regras ($O|\mathcal{C}|$). Como nesta abordagem a instanciação de restrições da aplicação é limitada por $O(|\mathcal{C}| \times |upd|)$, o tamanho de U é $O(|\mathcal{C}|^2 \times |upd|)$. A função $Clean(U)$ é $O(|\mathcal{C}|^2 \times |upd|)^2$. O Algoritmo AlgRDF mesmo quando trabalhado em um arquivo simples, uma versão não otimizada onde a tarefa mais custosa é limitada por $O(|U|^2 \times |D|^2)$. Portanto, sua complexidade é $O(|\mathcal{C}|^4 \times |upd|^2 \times |D|^2)$, uma vez que o tempo de execução do Algoritmo AlgTrad é o termo menos significativo.

5.5 DISCUSSÃO

Neste capítulo foi apresentado em detalhe o processo implementado pelo Sistema BNS. Também foram provadas formalmente a corretude e completude dos algoritmos. Na próxima seção são apresentados os experimentos que foram realizados utilizando a implementação destes algoritmos além de uma comparação com a proposta do Sistema FKAC [FKAC13].

6 ESTUDO EXPERIMENTAL

Este capítulo apresenta uma validação experimental do sistema BNS apresentado no Capítulo 5. A Seção 6.1 descreve a estratégia de comparação entre o sistema BNS e a abordagem FKAC [FKAC13]. Detalhes da implementação e dos *benchmarks* utilizados nos experimentos são apresentados na Seção 6.2. A Seção 6.3 descreve e analisa os resultados dos experimentos. Um estudo de caso utilizando dados reais da Companhia de Urbanização de Curitiba (URBS) é apresentado na Seção 6.4.

6.1 ESTRATÉGIA DE COMPARAÇÃO ENTRE OS SISTEMAS BNS E FKAC

Os Algoritmos do sistema BNS apresentados no Capítulo 5 foram implementados em SML# [OU11]. Para efeito de comparação, foi implementada também a abordagem proposta por [FKAC13], que aqui será chamada “sistema FKAC”. Foram feitas algumas modificações no sistema FKAC para que a comparação seja mais justa. A proposta FKAC original é baseada no princípio da alteração mínima. Para processar uma alteração, são computados todos os conjuntos possíveis de efeitos colaterais e realizada uma ordenação entre eles para selecionar aquele que proporciona a alteração mínima na base de dados. Neste processo, todos os resultados são considerados, incluindo aqueles que incluem alterações no esquema. No entanto, o sistema BNS não considera alterações de esquema, e portanto estas operações também não foram consideradas no sistema FKAC, limitando a quantidade de soluções e reduzindo seu tempo de execução.

Além disso, o sistema FKAC não considera a existência de nodos brancos na base ou tampouco o caracter especial “_” nas operações de atualização. Portanto, os experimentos foram conduzidos da seguinte forma. Em primeiro lugar, foram geradas as traduções de restrições de integridade do esquema RDF (Algoritmo AlgTrad do Capítulo 5) e estas restrições foram unidas às restrições de integridade da aplicação existentes. Posteriormente, foram gerados os efeitos colaterais U , baseados nas restrições da aplicação usando o Algoritmo AlgAplic. Então, antes de utilizar U no sistema FKAC, foram substituídas as operações com o caracter especial “_”. Para cada operação de remoção ($\neg PI(_, b, p)$ ou $\neg PI(a, _, p)$) a base de dados é acessada para substituir “_” por todos os átomos que devem ser removidos, de acordo com a semântica da operação definida na Tabela 4.2. Para cada operação de inserção ($PI(_, b, p)$ ou $PI(a, _, p)$) busca-se aleatoriamente uma entidade x do tipo requerido para a propriedade p para substituir o caracter “_”. Embora esta solução pareça arbitrária, pois cria uma relação $PI(x, b, p)$ ou $PI(a, x, p)$ não introduzida pelo usuário, ela está de acordo com a proposta do sistema FKAC. Se não há uma substituição possível, procede-se a remoção da entidade a (ou b).

Os experimentos foram conduzidos usando os Benchmarks Berlin [BS09] e LUBM [GPH05], como detalhado na próxima seção.

6.2 DESENVOLVIMENTO DOS SISTEMAS E BENCHMARKS UTILIZADOS

Inicialmente os experimentos foram conduzidos a partir de uma implementação em SML-NJ (Standard ML of New Jersey). Esta implementação considerava dados armazenados em arquivos, o que causava lentidão de processamento. Assim, houve uma adaptação

do código para SML#, para permitir que os dados pudessem ser armazenados no banco de dados PostgreSQL. Esta nova implementação melhorou o desempenho consideravelmente.

Foi criada uma tabela para cada um dos predicados existentes (CI, PI, Ind, Cl, Dom, Rng, Pr, Csub, Psub) efetuando junções conforme a necessidade de relacionar instâncias (CI) com indivíduos (Ind) ou relacionamentos (PI) com seu domínio (Dom) e imagem (Rng). Desta forma, recursos do SGBD como índices ou visões não foram usados o que ainda possibilita novas investidas no futuro para melhoria de desempenho no acesso à base de dados.

Os experimentos consideram dados de dois *benchmarks*: Berlin e LUBM. Para cada um foram identificadas as restrições de aplicação baseadas na documentação fornecida pelos desenvolvedores. Em seguida os dois sistemas foram executados, considerando 1, 5, 10, 20 e 40 inserções e igual número de remoções. Também foram considerados conjuntos com proporções variáveis de inserção/remoção: 70/30, 50/50 e 30/70. Como resultado, foram coletados os tempos de execução (sempre uma média de três execuções) e a quantidade de efeitos colaterais gerados.

6.2.1 O *Benchmark* Berlin

O *Benchmark* Berlin gera uma base de produtos com características, produtores, revisões (e suas publicações) e ofertas. Uma base de dados com 500 produtos foi gerada com 220 mil triplas. Assim, foi povoada uma base de dados com 900 mil fatos, de acordo com a notação apresentada na Seção 2.1. Na base armazenada em PostgreSQL, as 220 mil triplas foram armazenadas na tabela *PI*. Baseadas na especificação do *benchmark*, foram identificadas 10 restrições de aplicação que também foram traduzidas para a notação utilizada neste trabalho e detalhadas na Figura 6.1.

Todo produto tem um produtor	$C_1: CI(p, Produto) \rightarrow PI(p, a, ProduzidoPor)$
Todo produtor produz pelo menos um produto	$C_2: CI(a, Produtor) \rightarrow PI(p, a, ProduzidoPor)$
Toda revisão se refere a um produto	$C_3: PI(r, p, RevisãoDe) \rightarrow CI(p, Produto)$
	$C_4: PI(r, p, RevisãoDe) \rightarrow CI(r, Revisão)$
Toda oferta está associada um produto e um vendedor	$C_5: PI(p, v, Oferta) \rightarrow CI(v, Vendedor)$
	$C_6: PI(p, v, Oferta) \rightarrow CI(p, Produto)$
Todo revisor faz pelo menos uma revisão	$C_7: CI(a, Revisor) \rightarrow PI(a, b, Publica)$
Toda revisão tem um revisor	$C_8: CI(b, Revisão) \rightarrow PI(a, b, Publica)$
Toda característica está relacionada a um produto	$C_9: CI(a, Característica) \rightarrow PI(a, b, CaracProduto)$
Todo produto tem pelo menos uma característica	$C_{10}: CI(b, Produto) \rightarrow PI(a, b, CaracProduto)$

Figura 6.1: Restrições de Aplicação do Benchmark Berlin

Nos conjuntos de atualização foram utilizadas operações que ativassem diversas restrições e também operações que não causassem violação de nenhuma delas. O objetivo é que o conjunto fosse o mais neutro possível e não favorecesse nenhuma das estratégias. Os conjuntos de atualização utilizados estão detalhados no anexo A.1.

6.2.2 O Benchmark LUBM

O *Benchmark* LUBM gera uma base de dados universitários, com professores e alunos (e suas formações em nível de graduação e pós-graduação), disciplinas (ministradas e cursadas) e orientações. Uma base de dados foi gerada com 3 universidades e 14 departamentos cada, totalizando 198 mil triplas e um total de 1 milhão e 780 mil fatos. Baseado na especificação do *benchmark*, foram identificadas 10 restrições de aplicação que estão listadas na Figura 6.2 .

Todo professor trabalha em um departamento	$C_1: CI(a, Professor) \rightarrow PI(a, b, TrabalhaPara)$
Todo professor ministra uma disciplina	$C_2: CI(a, Professor) \rightarrow PI(a, b, Ministra)$
Todo aluno é membro de um departamento	$C_3: CI(a, Aluno) \rightarrow PI(a, b, MembroDe)$
Todo aluno de pós tem orientador	$C_4: CI(b, AlunoPos) \rightarrow PI(a, b, Orienta)$
Todo aluno cursa alguma disciplina	$C_5: CI(a, Aluno) \rightarrow PI(a, b, Cursa)$
Toda publicação tem autor	$C_6: CI(a, Publicação) \rightarrow PI(a, b, PublAutor)$
Todo professor tem formação	$C_7: CI(a, Professor) \rightarrow PI(a, b, GraduadoEm)$
	$C_8: CI(a, Professor) \rightarrow PI(a, b, MestradoEm)$
	$C_9: CI(a, Professor) \rightarrow PI(a, b, DoutoradoEm)$
Todo aluno de pós é graduado	$C_{10}: CI(a, AlunoPos) \rightarrow PI(a, b, GraduadoEm)$

Figura 6.2: Restrições de Aplicação do Benchmark LUBM

Da mesma forma que para o *benchmark* Berlin, os conjuntos de atualização consistem de operações que causam a violação de restrições bem como operações sobre as quais as restrições não tem efeito. Os conjuntos utilizados estão detalhados no anexo A.2.

6.3 RESULTADOS DOS EXPERIMENTOS

Esta seção detalha os resultados experimentais obtidos com os *benchmarks* Berlin e LUBM. Os tempos de execução reportados consistem da média coletada após três execuções em cada caso.

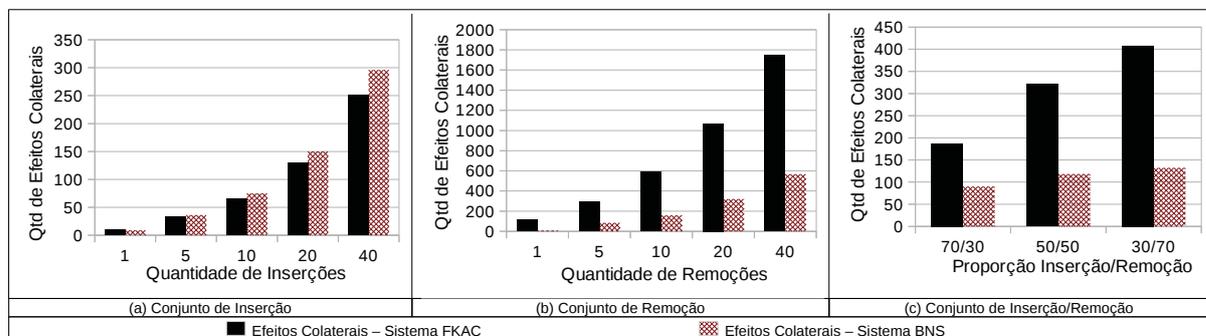


Figura 6.3: Quantidade de efeitos colaterais gerados com o benchmark Berlin com operações de atualização: somente inserção (a), somente remoção (b) e Misto (c)

As Figuras 6.3 e 6.4 mostram a quantidade de efeitos colaterais gerados para os *benchmarks* Berlin e LUBM, respectivamente. É possível notar que tanto para a inserção,

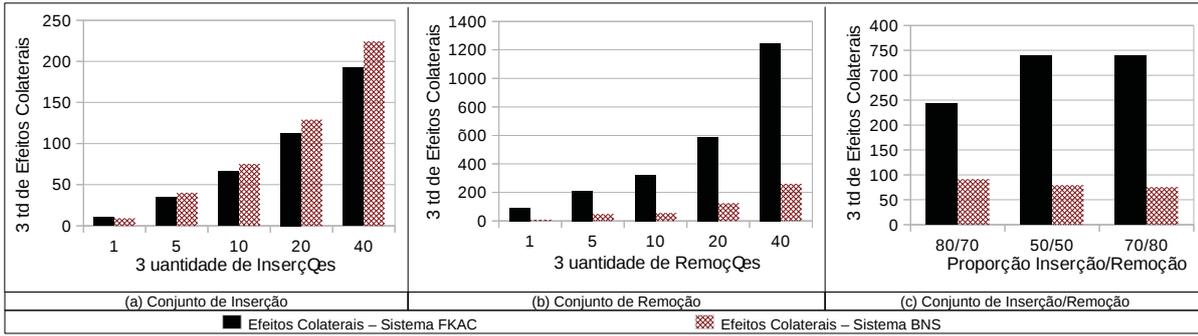


Figura 6.4: Quantidade de efeitos colaterais gerados com o benchmark LUBM com operações de atualização: somente inserção (a), somente remoção (b) e Misto (c)

a remoção ou em conjuntos com os dois tipos de operações, o comportamento do gráfico é semelhante para os dois *benchmarks*. Para operações de inserção a quantidade de efeitos colaterais gerados tende a ser maior para o sistema BNS, enquanto o sistema FKAC gera uma quantidade maior no caso de remoções e mistura de operações.

Isto ocorre quando o Algoritmo AlgAplic gera relacionamentos com nodos brancos. Como o sistema FKAC não prevê este tipo de nodo, é aplicada a solução proposta pelos autores de uma associação arbitrária com outro dado existente na base. Ou seja, ao invés da inserção de $PI(N_1, b, p)$ (ou $PI(a, N_1, p)$) realizada no sistema BNS, para o sistema FKAC é procurada uma entidade arbitrária x do tipo de domínio (ou imagem) da propriedade p e executada a inserção de $PI(x, b, p)$ (ou $PI(a, x, p)$). Enquanto isso, para o Sistema BNS a inserção de $PI(N_1, b, p)$ (ou $PI(a, N_1, p)$) também causa a inserção do nodo branco ($BN(N_1)$). Ou seja, existe uma operação adicional no sistema BNS a cada efeito colateral envolvendo nodos brancos.

No entanto, apesar da inserção adicional do nodo branco, a solução adotada pelo sistema BNS é semanticamente melhor. Por exemplo, considere que na base de dados existe um fato $PI(iPhone, Apple, ProduzidoPor)$. Caso este fato seja removido, haverá a violação da restrição C_1 e C_2 na Tabela 6.1. No sistema BNS haverá a inserção de $PI(iPhone, N_1, ProduzidoPor)$ enquanto no sistema FKAC haverá a busca por um dado com a mesma imagem para a propriedade $ProduzidoPor$ e poderá haver a inserção $PI(iPhone, Motorola, ProduzidoPor)$, o que nem sempre está de acordo com a realidade. Além disso, esta inserção arbitrária permanece na base até que seja explicitamente substituída pelo usuário. No caso do sistema BNS, quando houver a inserção de um real produtor x , ou seja $PI(iPhone, x, ProduzidoPor)$, a associação com o nodo branco N_1 é substituída e o nodo branco removido.

No caso de remoção, a quantidade de efeitos colaterais para o sistema FKAC é maior devido ao efeito cascata. Considere, por exemplo, no *benchmark* Berlin a atualização $\neg PI(a, b, ProduzidoPor)$, onde a é um produto específico e b é um produtor. Esta atualização gera um conjunto de efeitos colaterais devido ao uso de dois conjuntos de restrições, dentre as restrições identificadas: o primeiro requer que todo *Produto* tenha um *Produtor* (C_1 e C_2) e o segundo que toda característica de produto (*CaracProduto*) esteja associada a um *Produto* (C_9 e C_{10}).

Com este caso, o sistema FKAC gerou remoções em cascata, de um produto e suas características. No sistema BNS, por outro lado, o produtor exigido foi substituído por um nodo branco, interrompendo a necessidade de remoções em cascata.

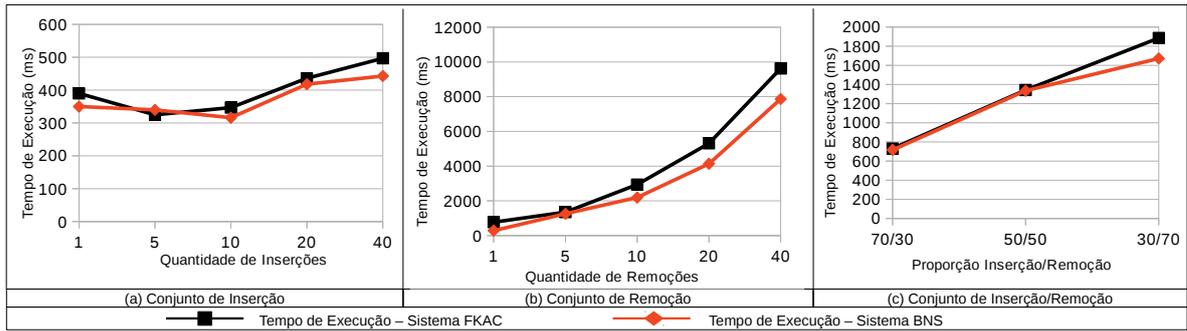


Figura 6.5: Tempo de Execução de atualizações no benchmark Berlin: Inserção (a), Remoção (b) e Misto (c)

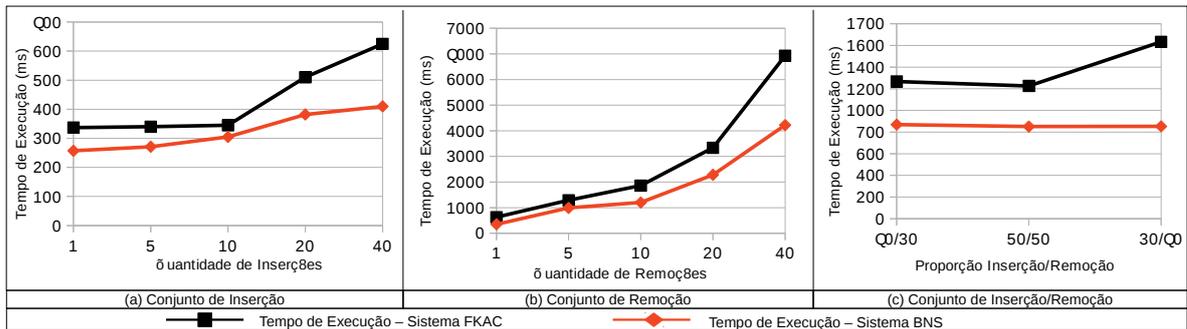


Figura 6.6: Tempo de Execução de atualizações no benchmark LUBM: Inserção (a), Remoção (b) e Misto (c)

As Figuras 6.5 e 6.6 mostram o tempo de execução para os dois benchmarks. Eles mostram que o tempo de execução é similar quando se tem o mesmo tamanho no conjunto de atualização, aumentando conforme se aumenta o conjunto, o que já era esperado.

Como no caso de remoção e da mistura de operações, o efeito cascata causa um maior número de efeitos colaterais, uma consequência natural é o maior tempo de execução para o sistema FKAC com as duas bases de dados utilizadas.

No caso de inserções, mesmo que a quantidade de efeitos colaterais seja menor, o tempo de execução foi maior para a estratégia FKAC. Isto se deve a necessidade de, em caso de relacionamento com um nodo branco, buscar na base de dados uma informação que possa substituí-lo por um dado já existente. Ainda que não seja uma operação excessivamente custosa, isso causa impacto dependendo da quantidade de nodos brancos gerados pelo conjunto de atualização. Por isto é possível verificar no gráfico que em vários casos o tempo de execução se aproxima muito entre os dois sistemas (quando a quantidade de nodos brancos é menor).

6.4 ESTUDO DE CASO

Um estudo de caso considerando uma base real da Companhia de Urbanização de Curitiba (URBS) acerca das linhas e horários de ônibus, foi utilizada para validação do sistema BNS.

Os dados da URBS estão armazenados originalmente em uma base de dados relacional. Atualmente existe na Companhia um esforço no sentido de tornar os dados do transporte coletivo disponíveis para toda a sociedade e uma possibilidade para publicar

estas informações é o formato RDF, devido a sua simplicidade e facilidade de interpretação por máquinas e humanos.

Partindo desta convergência entre o interesse da URBS e o sistema BNS, foram convertidos para o modelo RDF os dados do sistema de transporte considerando os dois principais símbolos do transporte coletivo de Curitiba: as linhas expressas (que utilizam corredores exclusivos e ligam diferentes regiões da cidade de maneira rápida) e suas plataformas de embarque rápido (as estações tubo e os terminais urbanos).

A base resultante contém propriedades de esquema tais como classes, subclasses, domínio, imagem, propriedades e subpropriedades, bem como triplas RDF. A Figura 6.7 ilustra como foi feita a conversão dos dados. Neste exemplo, a partir da tabela de linhas (e suas categorias) são identificadas linhas expressas (a) que então são associadas com as paradas para cada linha (b). Associando estas tabelas, é gerada a tabela RDF representada na Figura 6.7(c) com a criação da propriedade “*ExpressLineStop*” e associando cada parada para as linhas do tipo “Expresso”. Por exemplo, na tabela de rotas é possível identificar a linha 302 como sendo do tipo linha expressa. Na tabela de paradas, a “Estação Tubo Germânia” está listada como uma parada para a linha 302. Assim gera-se a tripla $PI(302, ExpressLineStop, Estacao\ Tubo\ Germania)$.

code character varying(25)	name character varying(100)	category character varying(100)	color character varying(100)
Z01	UNIBRASIL / TUBO DET	CONVENCIONAL	AMARELA
203	STA. CÂNDIDA / C. RA	EXPRESSO	VERMELHA
302	CENTENÁRIO / RUI BAR	EXPRESSO	VERMELHA
303	CENTENÁRIO / C. COMP	EXPRESSO	VERMELHA

(a)

line_code character varying(25)	address character varying(254)
302	Estação Tubo Antônio Meirelles
302	Estação Tubo Urbano Lopes
302	Terminal Oficinas - 303 - Cente
302	Estação Tubo Teófilo Otoni
302	Estação Tubo Del. Amazor Preste
302	Estação Tubo Antônio Meirelles
302	Estação Tubo Cajuru
302	Estação Tubo Hospital Cajuru

(b)

subject character varying(200)	predicate character varying(200)	object character varying(200)
302	ExpressLineStop	Estação Tubo Praça Eufrasio Correia
302	ExpressLineStop	Estação Tubo Germânia
302	ExpressLineStop	Estação Tubo Profª. Maria Aguiar Teixeir
302	ExpressLineStop	Estação Tubo Profª. Maria Aguiar Teixeir
302	ExpressLineStop	Estação Tubo Praça Eufrasio Correia
302	ExpressLineStop	Estação Tubo Jardim Botânico
302	ExpressLineStop	Terminal Centenário - 303 - Centenário /
302	ExpressLineStop	Estação Tubo Profª. Maria Aguiar Teixeir
302	ExpressLineStop	Estação Tubo Catulo da P. Cearense
302	ExpressLineStop	Estação Tubo Cajuru

(c)

Figura 6.7: Dados da Base de Dados Relacional da URBS e as Tripas Geradas

A partir deste processo foram geradas 110 mil triplas na tabela *PI*. Foram ainda identificadas algumas restrições de integridade para este sistema, conforme apresentado na Figura 6.8.

O objetivo de utilizar o sistema BNS em conjunto com a base da URBS, além da validação das restrições de integridade é auxiliar ao processo de atualização da base de dados. Suponha que existam 3 tipos de paradas de ônibus: terminais, tubos e pontos comuns e que apenas os dois primeiros sejam consideradas de embarque rápido. A partir disto, considere uma atualização, inserindo uma parada “s1” para a linha 302. De acordo com a restrição c_4 da Figura 6.8 “s1” deve ser uma parada de embarque rápido e portanto não pode ser um ponto comum. No entanto, se “s1” já está armazenado na base de dados como um ponto comum existem duas alternativas: (i) a atualização gera uma inconsistência na base de dados e portanto deve ser rejeitada; ou (ii) a base de dados é inconsistente e “s1” deve ser inserida como uma parada de embarque rápido. Como o sistema BNS detecta

Pagamento em microônibus é feito apenas com cartão	$C_1: CI(v, MicroOnibus) \rightarrow CI(v, ApenasCartão)$
Toda linha expressa tem ponto inicial	$C_2: CI(l, LinhaExpressa) \rightarrow PI(s, l, PontoInicial)$
Toda linha expressa tem ponto final	$C_3: CI(l, LinhaExpressa) \rightarrow PI(s, l, PontoFinal)$
Toda parada de linha expressa é de embarque rápido	$C_4: PI(s, l, ParadaExpresso) \rightarrow CI(s, EmbarqueRapido)$
Toda estação tubo é de embarque rápido	$C_5: CI(s, Tubo) \rightarrow CI(s, EmbarqueRapido)$
Todo terminal é de embarque rápido	$C_6: CI(s, Terminal) \rightarrow CI(s, EmbarqueRapido)$
Paradas de rua não são de embarque rápido	$C_7: CI(s, PontoRua) \rightarrow \neg CI(s, EmbarqueRapido)$
Linhas expressas tem paradas e veículos específicos	$C_8: PI(s, l, ParadaLinhaExpr) \rightarrow CI(l, LinhaExpressa)$
	$C_9: PI(s, l, VeiculoExpresso) \rightarrow CI(l, LinhaExpressa)$
	$C_{10}: PI(s, l, ParadaLinhaExpr) \rightarrow CI(s, ParadaExpresso)$

Figura 6.8: Restrições de Aplicação do sistema URBS

estas inconsistências, ele pode ser utilizado como apoio pelo usuário para que ele aceite as atualizações (e neste caso corrija o tipo da parada “s1”) ou então rejeite a atualização (por detectar que ela não está de acordo com dados já existentes na base).

O resultado dos experimentos foi muito semelhante àquele com os *benchmarks*, como pode ser observado na Figura 6.9. Ou seja, para a inserção (a) a quantidade de efeitos colaterais gerada pelo sistema FKAC é menor em função da associação arbitrária mais simples e direta, ao passo que o sistema BNS possui uma semântica mais adequada.

Na remoção (b) e nos conjuntos mistos (c), o tempo de execução é menor para o sistema BNS em função da necessidade de remoções em cascata para o sistema FKAC.

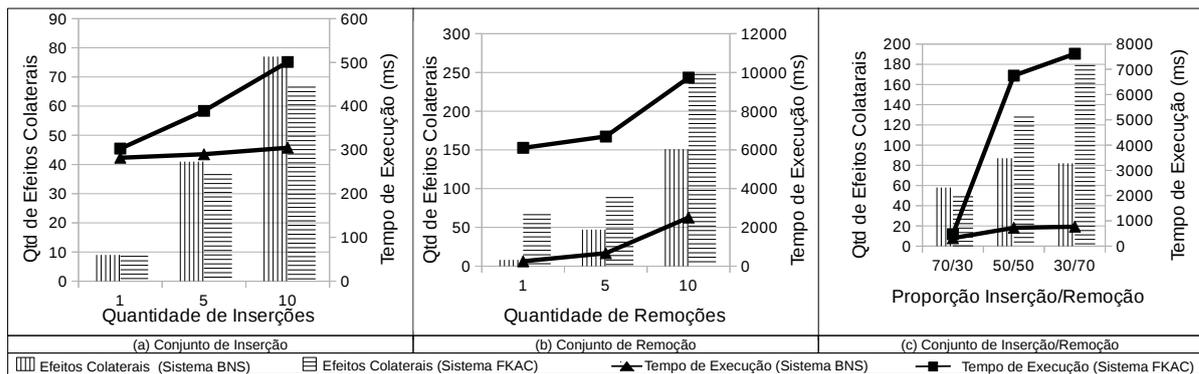


Figura 6.9: Resultado Obtido na Execução dos Sistemas para a base da URBS

6.5 DISCUSSÃO

Os experimentos realizados demonstram que a interrupção das operações em cascata com uso de nodos brancos causa uma diminuição no número de operações geradas em virtude dos efeitos colaterais. Mesmo no caso da inserção em que o número de efeitos colaterais é maior, o tempo de execução também é menor para o Sistema BNS. Além disso, a abordagem de associação arbitrária no lugar de nodos brancos adotada pelo sistema FKAC, resulta em

uma quantidade menor de efeitos colaterais para a inserção. Por outro lado, ela causa um problema semântico, uma vez que tal aleatoriedade na escolha de uma entidade para se fazer a associação pode não condizer com a realidade. Um agravante que deve ser adicionado a isto é o fato de posteriormente tal associação não ser removida.

Por outro lado, a associação com um nodo branco, no Sistema BNS é removida tão logo o dado obrigatório, de acordo com as restrições de integridade, é inserido, conforme preconizado pela semântica definida na Tabela 4.2.

Fazendo uma analogia com a semântica de operações de remoção no modelo relacional, a abordagem adotada pelo sistema FKAC é similar ao *on delete cascade* enquanto a proposta deste trabalho adota a semântica *on delete set null*.

7 CONCLUSÃO

Esta tese apresenta uma estratégia de garantia de integridade em bases RDF com a utilização de nodos brancos. O objetivo é que estes nodos evitem operações em cascata e evitem a geração de operações arbitrárias que produzam resultados semanticamente questionáveis. Esta estratégia foi implementada no sistema BNS para verificação de sua eficácia.

No Capítulo 2 são apresentados conceitos básicos para o desenvolvimento do trabalho. Além de definições relacionados a RDF e nodos brancos, é importante destacar a distinção entre esquema e instância adotada nesta tese para bases RDF. Ainda que esta distinção não seja clara em diversas bases, esta divisão é necessária para o funcionamento do sistema BNS.

No Capítulo 3 são discutidas diversas estratégias que tratam restrições de integridade e nodos brancos em diferentes contextos, mas que estão, de alguma forma, relacionados ao trabalho desta tese. Assim, é possível compreender de que forma estes temas costumam ser desenvolvidos em outros trabalhos existentes na literatura e avaliar as semelhanças e diferenças com o sistema BNS.

No Capítulo 4 é tratado um outro conceito importante para o trabalho: as restrições de integridade. Este tema é apresentado separadamente dos demais conceitos por abordar diversas adaptações e definições utilizadas especificamente neste trabalho.

O Capítulo 5 apresenta os detalhes do sistema BNS e os algoritmos que compõem a ferramenta. Também são apresentadas as provas de corretude dos algoritmos e detalhes de como o sistema aborda as questões de implementação.

O Capítulo 6 descreve os experimentos utilizados a partir de dois *benchmarks* relevantes quando se trata de dados em formatos RDF: o LUBM e o BSBM (Berlin). Além disso, um estudo de caso considerando uma base de dados reais da Companhia de Urbanização de Curitiba (URBS) apresenta resultados similares aos anteriores demonstrando a viabilidade e a eficiência do Sistema BNS.

A Seção 7.1 apresenta uma avaliação do trabalho, enquanto na Seção 7.2 são apresentadas algumas possibilidades de continuidade da estratégia apresentada nesta tese. Por fim, a Seção 7.3 relaciona os trabalhos publicados em eventos internacionais com os resultados obtidos.

7.1 AVALIAÇÃO DO TRABALHO

A realização deste trabalho envolveu desafios relacionados sobretudo à implementação e a formalização de provas de corretude para os algoritmos. A utilização da linguagem SML e o seu casamento de padrões facilitou a busca de dados e a validação de operações de atualização em um primeiro momento. No entanto, para integrar a implementação com a utilização de um SGBD envolveu dificuldades relacionadas ao uso de um dialeto da linguagem, denominado SML#, ainda incipiente e com diversas limitações de uso e compreensão.

Além disso, todos os algoritmos tiveram a sua corretude provada e isso envolveu uma dificuldade adicional, uma vez que lidar com todo o formalismo necessário para as provas não é uma tarefa trivial.

Depois de demonstrada a viabilidade da estratégia através de experimentos, é possível ter uma visão geral do processo de desenvolvimento e de possibilidades de utilização da mesma. Em primeiro lugar, a estreita relação com a estratégia proposta por [FKAC13], sugere uma natural integração entre as duas estratégias. A limitação desta tese em considerar apenas operações de esquema poderia ser suprida pela estratégia FKAC ao passo que as associações arbitrárias utilizadas sobretudo na inserção ou operações em cascata, utilizadas em remoções poderiam ser eliminadas com a utilização de nodos brancos.

Uma implementação complementar poderia permitir que o usuário escolhesse a utilização de uma das estratégias para situações específicas que atendessem melhor suas necessidades. Para isto, haveria a necessidade de avaliar alterações em operações de esquema, para que elas passem a considerar operações com nodos brancos que seriam inseridas nas operações de instância. Em outras palavras, é preciso avaliar se a existência de nodos brancos causaria impacto no procedimento de atualização das operações de esquema proposta por [FKAC13].

Adicionalmente, a implementação da técnica deve ser testada considerando dados armazenados em uma *triple store*, para que os experimentos pudessem considerar o armazenamento em triplas diretamente, e não em bases relacionais como no caso dos experimentos aqui apresentados. Isto aproximaria os resultados com situações práticas de manipulação de bases RDF.

7.2 TRABALHOS FUTUROS

Alguns desafios ainda precisam ser enfrentados, como a necessidade de identificar nodos brancos que possam se referir a uma mesma entidade e unificá-los. Esta preocupação está presente em alguns dos trabalhos relacionados ao controle de versão, apresentados no Capítulo 3.

Uma possibilidade é que isto seja feito partindo da identificação de domínio e imagem associados ao relacionamento entre entidades e uma análise das entidades envolvidas para que a similaridade entre nodos brancos possa ser identificada.

Outro desafio é expandir a estratégia para que ela passe a considerar operações de atualização envolvendo esquema (operações com predicados em SCHPRED). Este tipo de alteração é mais complexo porque pode afetar outras instâncias da base de dados e em alguns casos é possível que alterações nestas instâncias sejam necessárias para garantir a integridade da base.

No caso da inserção de novas classes, propriedades, domínios e imagens as consequências são mínimas já que não há instâncias associadas a eles e as novas instâncias já atenderão ao novo esquema estabelecido.

Para o caso de alterações no domínio e na imagem de propriedades já existentes, ou seja o conjunto *upd* ter dentre as suas operações $\neg Dom(x, y)$ e $Dom(x, z)$ é preciso verificar se tais alterações violam alguma regra da aplicação em relação ao domínio da propriedade x .

A atribuição de subclasses e subpropriedades entre entidades já existentes tem o mesmo problema. Como serão inseridas as instâncias de classe/propriedade equivalentes, é preciso verificar as regras da aplicação para evitar que isso cause inconsistências.

Caso ocorram inconsistências, será preciso uma análise para determinar a possibilidade de propor uma série de alterações na base para resolvê-las ou identificar a impossibilidade de executar a alteração, caso existam conflitos entre as regras da aplicação.

Conforme já citado no Capítulo 3, a utilização de tecnologias como ShEx [Sh14] e SHACL [KR17] em conjunto com o sistema BNS para validação de grafos RDF é outra atividade que merece atenção no futuro.

Outra atividade que deve concentrar os primeiros esforços, é a possibilidade de utilizar as restrições de integridade para obter o esquema de uma base RDF. A partir de uma restrição do tipo $PI(a, b, coordena) \rightarrow CI(a, Pesquisador)$ é possível deduzir que o domínio da propriedade coordena é *Pesquisador*. Além disso, restrições do tipo $CI(a, Pesquisador) \rightarrow CI(a, Professor)$ podem ser utilizadas para identificar subconjuntos entre instâncias e assim deduzir relações de subclasses. Estes são apenas alguns exemplos que mostram a riqueza de informações que podem ser deduzidas a partir de restrições de integridade.

Alguns experimentos iniciais já foram desenvolvidos neste sentido e a formalização deste processo pode ser utilizado em conjunto com estratégias de extração de esquema, otimizando seus resultados.

7.3 PUBLICAÇÕES COM RESULTADOS DA TESE

Halfeld-Ferrari, M., Hara, C. S. e Uber, F. R. (2017). RDF Updates with Constraints. Em KESW 2017 - International Conference on Knowledge Engineering and the Semantic Web. 2017. Páginas 229 - 245

Halfeld-Ferrari, M. Hara, C. S., Koziévitch, N. P. e Uber, F. R. (2018). Urban Data Consistency in RDF: A Case Study of Curitiba Transportation System. Em Latin American Workshop on Data Science (LADaS) - a VLDB Workshop. 2018.

REFERÊNCIAS

- [ACP10] Waseem Akhtar, Alvaro Cortés-Calabuig, and Jan Paredaens. Constraints in RDF. In *Proc of the 4th Int Work on Semantics in Data and Knowledge Bases*, pages 23–39, 2010.
- [AH06] Sören Auer and Heinrich Herre. A versioning and evolution framework for RDF knowledge bases. In *Proc of at 6th Int Conf on Perspectives of System Informatics*, 2006.
- [BS09] Christian Bizer and Andreas Schultz. The berlin sparql benchmark. *International Journal On Semantic Web and Information Systems*, 5(2):1–24, 2009.
- [CGK08] Andrea Cali, Georg Gottlob, and Michael Kifer. Taming the infinite chase: Query answering under expressive relational constraints. In *Proc of the 11th Int Conf on Principles of Knowledge Repres and Reasoning*, pages 70–80, 2008.
- [FES⁺16] Sidra Faisal, Kemele M. Endris, Saeedeh Shekarpour, Sören Auer, and Maria-Esther Vidal. Co-evolution of rdf datasets. In Alessandro Bozzon, Philippe Cudre-Maroux, and Cesare Pautasso, editors, *Web Engineering: 16th International Conference, ICWE 2016, Lugano, Switzerland, June 6-9, 2016. Proceedings*, pages 225–243, Cham, 2016. Springer International Publishing.
- [FKAC13] Giorgos Flouris, George Konstantinidis, Grigoris Antoniou, and Vassilis Christophides. Formal foundations for RDF/S KB evolution. *Knowl. Inf. Syst.*, 35(1):153–191, 2013.
- [FKP05] Ronald Fagin, Phokion G. Kolaitis, and Lucian Popa. Data exchange: getting to the core. *ACM Trans. Database Syst.*, 30(1):174–210, 2005.
- [FL17] Wenfei Fan and Ping Lu. Dependencies for graphs. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS '17, pages 403–416, New York, NY, USA, 2017. ACM.
- [FPA⁺16] Marvin Frommhold, Rubén Navarro Piris, Natanael Arndt, Sebastian Tramp, Niklas Petersen, and Michael Martin. Towards Versioning of Arbitrary RDF Data. In *Proc of the 12th Int Conf on Semantic Systems Proceedings*, 2016.
- [GPH05] Yuanbo Guo, Zhengxiang Pan, and Jeff Heffin. Lubm: A benchmark for owl knowledge base systems. *Web Semant.*, 3(2-3):158–182, 2005.
- [HAMP14] Aidan Hogan, Marcelo Arenas, Alejandro Mallea, and Axel Polleres. Everything you always wanted to know about blank nodes. *J. Web Sem.*, 27:42–69, 2014.
- [HAR11] Jiewen Huang, Daniel J. Abadi, and Kun Ren. Scalable sparql quering of large rdf graphs. In *VLDB Endowment*, 2011.

- [HFL17] Mirian Halfeld-Ferrari and Dominique Laurent. Updating rdf/s databases under constraints. In Mārīte Kirikova, Kjetil Nørnvåg, and George A. Papadopoulos, editors, *Advances in Databases and Information Systems*, pages 357–371, Cham, 2017. Springer International Publishing.
- [HLS98] Mirian Halfeld-Ferrari, Dominique Laurent, and Nicolas Spyratos. Update rules in datalog programs. *J. Log. Comput.*, 8(6):745–775, 1998.
- [Hog15] Aidan Hogan. Skolemising blank nodes while preserving isomorphism. In *Proceedings of the 24th International Conference on World Wide Web, WWW '15*, pages 430–440, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.
- [KR17] H. Knublauch and A. Ryman. Shapes constraint language (SHACL). W3C first public working draft, w3c. <http://www.w3.org/TR/2015/WD-shacl-20151008/>, 2017.
- [MAHP11] Alejandro Mallea, Marcelo Arenas, Aidan Hogan, and Axel Polleres. On blank nodes. In *Proc of the 10th Int Semantic Web Conference*, pages 421–437, 2011.
- [Mn16] Emir Muñoz. On learnability of constraints from RDF data. In *Proc. of the 13th Extended Semantic Web Conference*, pages 834–844, 2016.
- [MSCK05] Matoula Magiridou, S. Sahtouris, Vassilis Christophides, and Manolis Koubarakis. RUL: A declarative update language for RDF. In *Proc of the 4th Int Semantic Web Conference*, pages 506–521, 2005.
- [OU11] Atsushi Ohori and Katsuhiko Ueno. Making standard ml a practical database programming language. *SIGPLAN Not.*, 46(9):307–319, September 2011.
- [SG13] Alessandro Solimando and Giovanna Guerrini. Ontology adaptation upon updates. In *ESWC Satellite Events*, pages 34–45, 2013.
- [Sh14] H. Solbrig and E. Prud'hommeaux. Shape expressions 1.0 definition. W3C member submission. <http://www.w3.org/Submission/2014/SUBM-shex-defn-20140602>, 2014.
- [VG06] Max Völkel and Tudor Groza. SemVersion: An RDF-based Ontology Versioning System. In Miguel Baptista Nunes, editor, *Proc of IADIS International Conf on WWW/Internet*, pages 195–202, 2006.
- [YHLX18] L. Yang, L. Huang, H. Lu, and F. Xu. Removal mechanism of redundant blank nodes in linked data. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 1118–1121, May 2018.

APÊNDICE A – CONJUNTO DE ATUALIZAÇÕES

Para os estudos experimentais apresentados no Capítulo 6 foram utilizadas três bases de dados: uma para o benchmark Berlin, outra para o benchmark LUBM e ainda a base de dados da URBS para o estudo de caso. A seguir serão apresentados os conjuntos de atualizações utilizados para cada uma delas. É importante lembrar que tanto para o Sistema BNS quanto para o Sistema FKAC foi submetido o mesmo conjunto de atualizações.

A.1 BENCHMARK BERLIN

Conjunto de atualizações:

- 1 inserção: upd =
 $\{CI(\langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product1234} \rangle, \langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product} \rangle)\}$
- 5 inserções: upd =
 $\{CI(\langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product1234} \rangle, \langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product} \rangle),$
 $CI(\langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product1245} \rangle, \langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product} \rangle),$
 $PI(\langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product1234} \rangle, \langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Producer2} \rangle, \text{“publisher”}),$
 $PI(\langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product1245} \rangle, \langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Producer3} \rangle, \text{“publisher”}),$
 $PI(\langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product3} \rangle, \langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/ProductFeature600} \rangle, \text{“productFeature”})\}$
- 10 inserções: upd =
 $\{CI(\langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product1234} \rangle, \langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product} \rangle),$
 $CI(\langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product1245} \rangle, \langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product} \rangle),$
 $PI(\langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product1234} \rangle, \langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Producer2} \rangle, \text{“publisher”}),$
 $PI(\langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product1245} \rangle, \langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Producer3} \rangle, \text{“publisher”}),$
 $PI(\langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product3} \rangle, \langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/ProductFeature600} \rangle, \text{“productFeature”}),$
 $CI(\langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product3344} \rangle, \langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product} \rangle),$
 $CI(\langle \text{http} : // \text{www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/}$


```

PI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product3636 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/
dataFromProducer1/Producer6 > ", "publisher"),
PI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product8 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/
instances/ProductFeature890 > ", "productFeature"),
CI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product8484 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
CI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product9393 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
PI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product4747 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/
dataFromProducer1/Producer8822 > ", "publisher"),
PI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product5588 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/
dataFromProducer1/Producer9922 > ", "publisher"),
PI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product3 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/
instances/ProductFeature430 > ", "productFeature"))}

```

- 1 remoção: upd =

```

{NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/
dataFromProducer1/Producer1 > ", "publisher")}

```
- 5 remoções: upd =

```

{NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/
dataFromProducer1/Producer1 > ", "publisher"),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/
Product118 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/
instances/ProductFeature683 > ", "productFeature"),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product2 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/
Offer73 > ", "2008-05-14T00:00:00|http://www.w3.org/2001/
XMLSchemadateTime > ", "validFrom"),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/
Offer33 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Offer > ")}

```
- 10 remoções: upd =

```

{NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/
dataFromProducer1/Producer1 > ", "publisher"),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/
Product118 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/
instances/ProductFeature683 > ", "productFeature"),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product2 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/

```

```

Product > "),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/
Offer73 > ", "2008-05-14T00:00:00|http://www.w3.org/2001/
XMLSchemadateTime > ", "validFrom"),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/
Offer33 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Offer > "),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/
Product118 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/
dataFromProducer3/Producer3 > ", "publisher"),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/
Product11 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/
instances/ProductFeature34 > ", "productFeature"),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product3 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product4 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/
Offer23 > ", "2008-05-14T00:00:00|http://www.w3.org/2001/
XMLSchemadateTime > ", "validFrom"),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/
Offer43 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Offer > ")}

```

- 20 remoções: upd =

```

{NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/
dataFromProducer1/Producer1 > ", "publisher"),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/
Product118 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/
instances/ProductFeature683 > ", "productFeature"),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product2 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/
Offer73 > ", "2008-05-14T00:00:00|http://www.w3.org/2001/
XMLSchemadateTime > ", "validFrom"),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/
Offer33 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Offer > "),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/
Product118 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/
dataFromProducer3/Producer3 > ", "publisher"),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/
Product11 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/
instances/ProductFeature34 > ", "productFeature"),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product3 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product4 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/

```

Offer23 > ", "2008 - 05 - 14T00 : 00 : 00|<http://www.w3.org/2001/XMLSchema#dateTime> > ", "validFrom"),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer43> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Offer> > "),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product10> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Producer1> > ", "publisher"),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Product45> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/ProductFeature32> > ", "productFeature"),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product5> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product6> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer8> > ", "2008 - 05 - 14T00 : 00 : 00|<http://www.w3.org/2001/XMLSchema#dateTime> > ", "validFrom"),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer22> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Offer> > "),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Product23> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Producer3> > ", "publisher"),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Product11> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/ProductFeature10> > ", "productFeature"),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product7> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product8> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer99> > ", "2008 - 05 - 14T00 : 00 : 00|<http://www.w3.org/2001/XMLSchema#dateTime> > ", "validFrom"),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer11> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Offer> > ") }

- 40 remoções: upd =

{i*NPI*(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product1> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Producer1> > ", "publisher"),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Product118> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/instances/ProductFeature683> > ", "productFeature"),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product1> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product2> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer73> > ", "2008 - 05 - 14T00 : 00 : 00|<http://www.w3.org/2001/XMLSchema#dateTime> > ", "validFrom"),

NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer33> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Offer> > "),
 NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Product118> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Producer3> > ", "publisher"),
 NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Product11> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/ProductFeature34> > ", "productFeature"),
 NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product3> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
 NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product4> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
 NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer23> > ", "2008 - 05 - 14T00 : 00 : 00|<http://www.w3.org/2001/XMLSchemadateTime> > ", "validFrom"),
 NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer43> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Offer> > "),
 NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product10> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Producer1> > ", "publisher"),
 NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Product45> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/ProductFeature32> > ", "productFeature"),
 NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product5> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
 NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product6> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
 NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer8> > ", "2008 - 05 - 14T00 : 00 : 00|<http://www.w3.org/2001/XMLSchemadateTime> > ", "validFrom"),
 NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer22> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Offer> > "),
 NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Product23> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Producer3> > ", "publisher"),
 NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Product11> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/ProductFeature10> > ", "productFeature"),
 NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product7> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
 NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product8> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
 NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer99> > ", "2008 - 05 - 14T00 : 00 : 00|<http://www.w3.org/2001/XMLSchemadateTime> > ", "validFrom"),
 NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer11> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Offer> > "),
 NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/>

Product8 > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Producer1> > ", "publisher"),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Product58> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/ProductFeature33> > ", "productFeature"),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product9> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product10> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer72> > ", "2008-05-14T00:00:00|<http://www.w3.org/2001/XMLSchemadateTime> > ", "validFrom"),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer66> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Offer> > "),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Product43> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Producer3> > ", "publisher"),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Product11> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/ProductFeature88> > ", "productFeature"),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product11> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product12> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer55> > ", "2008-05-14T00:00:00|<http://www.w3.org/2001/XMLSchemadateTime> > ", "validFrom"),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer99> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Offer> > "),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product18> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Producer1> > ", "publisher"),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Product33> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/ProductFeature0> > ", "productFeature"),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product13> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/Product14> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Product> > "),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer44> > ", "2008-05-14T00:00:00|<http://www.w3.org/2001/XMLSchemadateTime> > ", "validFrom"),
NCI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/Offer90> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/Offer> > "),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Product21> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Producer1> > ", "publisher"),
NPI(" < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/Product234> > ", " < <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/>

```

instances/ProductFeature212 > ", "productFeature"),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product15 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product16 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/
Offer123 > ", "2008-05-14T00:00:00|http://www.w3.org/2001/
XMLSchemadateTime > ", "validFrom"),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/
Offer453 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Offer > ")

```

- 7 inserções / 3 remoções: upd =

```

{CI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product4455 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
PI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product8888 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/
dataFromProducer1/Producer8800 > ", "publisher"),
CI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1234 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
CI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1245 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
PI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1234 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/
dataFromProducer1/Producer2 > ", "publisher"),
PI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1245 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/
dataFromProducer1/Producer3 > ", "publisher"),
PI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product3 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/
instances/ProductFeature600 > ", "productFeature"),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/
instances/dataFromProducer1/Producer1 > ", "publisher"),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/
Product118 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/
instances/ProductFeature683 > ", "productFeature"),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > ")
}

```

- 5 inserções / 5 remoções: upd =

```

{CI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1234 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
CI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1245 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
PI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1234 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/
dataFromProducer1/Producer2 > ", "publisher"),
PI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1245 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/
dataFromProducer1/Producer3 > ", "publisher"),
}

```

```

PI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product3 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/
instances/ProductFeature600 > ", "productFeature"),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/
dataFromProducer1/Producer1 > ", "publisher"),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/
Product118 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/
instances/ProductFeature683 > ", "productFeature"),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product2 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/
Offer73 > ", "2008-05-14T00:00:00|http://www.w3.org/2001/
XMLSchemadateTime > ", "validFrom"),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/
Offer33 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Offer > ")
}

```

- 3 inserções / 7 remoções: upd =

```

{CI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1234 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
CI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1245 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
PI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1234 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/
dataFromProducer1/Producer2 > ", "publisher"),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/
dataFromProducer1/Producer1 > ", "publisher"),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer3/
Product118 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/
instances/ProductFeature683 > ", "productFeature"),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product1 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product2 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/
Offer73 > ", "2008-05-14T00:00:00|http://www.w3.org/2001/
XMLSchemadateTime > ", "validFrom"),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/
Offer33 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Offer > "),
NCI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromProducer1/
Product8 > ", " < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/vocabulary/
Product > "),
NPI(" < http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromVendor1/
Offer99 > ", "2008-05-14T00:00:00|http://www.w3.org/2001/
XMLSchemadateTime > ", "validFrom")}

```

A.2 BENCHMARK LUBM

Conjunto de atualizações:

- 1 inserção: upd =
`{CI("http://www.Department0.University0.edu/FullProfessor100", "Faculty")}`
- 5 inserções: upd =
`{CI("http://www.Department0.University0.edu/FullProfessor100", "Faculty"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication0",
"http://www.Department0.University0.edu/FullProfessor100", "publicationAuthor"),
CI("http://www.Department0.University0.edu/FullProfessor110", "Faculty"),
CI("http://www.Department12.University0.edu/UndergraduateStudent800", "Student"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication110",
"http://www.Department0.University0.edu/FullProfessor0", "publicationAuthor")}`
- 10 inserções: upd =
`{CI("http://www.Department0.University0.edu/FullProfessor100", "Faculty"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication0",
"http://www.Department0.University0.edu/FullProfessor100", "publicationAuthor"),
CI("http://www.Department0.University0.edu/FullProfessor110", "Faculty"),
CI("http://www.Department12.University0.edu/UndergraduateStudent800", "Student"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication110",
"http://www.Department0.University0.edu/FullProfessor0", "publicationAuthor"),
CI("http://www.Department14.University1.edu/GraduateStudent560", "student"),
PI("http://www.Department14.University1.edu/GraduateStudent650",
"http://www.Department14.University1.edu/AssociateProfessor4", "advisor"),
CI("http://www.Department14.University1.edu/FullProfessor0/Publication510",
"Publication"),
PI("http://www.Department0.University0.edu/FullProfessor12",
"http://www.Department0.University0.edu/Course0", "teacherOf"),
PI("http://www.Department6.University0.edu/UndergraduateStudent298",
"http://www.Department6.University0.edu/Course0", "takesCourse")}`
- 20 inserções: upd =
`{CI("http://www.Department0.University0.edu/FullProfessor100", "Faculty"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication0",
"http://www.Department0.University0.edu/FullProfessor100", "publicationAuthor"),
CI("http://www.Department0.University0.edu/FullProfessor110", "Faculty"),
CI("http://www.Department12.University0.edu/UndergraduateStudent800", "Student"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication110",
"http://www.Department0.University0.edu/FullProfessor0", "publicationAuthor"),
CI("http://www.Department14.University1.edu/GraduateStudent560", "student"),
PI("http://www.Department14.University1.edu/GraduateStudent650",
"http://www.Department14.University1.edu/AssociateProfessor4", "advisor"),
CI("http://www.Department14.University1.edu/FullProfessor0/Publication510",
"Publication"),
PI("http://www.Department0.University0.edu/FullProfessor12",
"http://www.Department0.University0.edu/Course0", "teacherOf"),
PI("http://www.Department6.University0.edu/UndergraduateStudent298",
"http://www.Department6.University0.edu/Course0", "takesCourse"),
CI("http://www.Department0.University0.edu/FullProfessor200",
"Faculty"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication0",
"http://www.Department0.University0.edu/FullProfessor200", "publicationAuthor"),
CI("http://www.Department1.University0.edu/FullProfessor210",
"Faculty"),
CI("http://www.Department12.University0.edu/UndergraduateStudent900",
"Student")}`

```

PI("http://www.Department0.University0.edu/FullProfessor0/Publication210",
"http://www.Department0.University0.edu/FullProfessor3", "publicationAuthor"),
CI("http://www.Department14.University1.edu/GraduateStudent750",
"student"),
PI("http://www.Department14.University1.edu/GraduateStudent750",
"http://www.Department14.University1.edu/AssociateProfessor1", "advisor"),
CI("http://www.Department14.University1.edu/FullProfessor0/Publication610",
"Publication"),
PI("http://www.Department0.University0.edu/FullProfessor3",
"http://www.Department0.University0.edu/Course1", "teacherOf"),
PI("http://www.Department6.University0.edu/UndergraduateStudent54",
"http://www.Department6.University0.edu/Course1", "takesCourse")}

```

- 40 inserções: upd =

```

{CI("http://www.Department0.University0.edu/FullProfessor100", "Faculty"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication0",
"http://www.Department0.University0.edu/FullProfessor100", "publicationAuthor"),
CI("http://www.Department0.University0.edu/FullProfessor110", "Faculty"),
CI("http://www.Department12.University0.edu/UndergraduateStudent800", "Student"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication110",
"http://www.Department0.University0.edu/FullProfessor0", "publicationAuthor"),
CI("http://www.Department14.University1.edu/GraduateStudent560", "student"),
PI("http://www.Department14.University1.edu/GraduateStudent650",
"http://www.Department14.University1.edu/AssociateProfessor4", "advisor"),
CI("http://www.Department14.University1.edu/FullProfessor0/Publication510",
"Publication"),
PI("http://www.Department0.University0.edu/FullProfessor12",
"http://www.Department0.University0.edu/Course0", "teacherOf"),
PI("http://www.Department6.University0.edu/UndergraduateStudent298",
"http://www.Department6.University0.edu/Course0", "takesCourse"),
CI("http://www.Department0.University0.edu/FullProfessor200",
"Faculty"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication0",
"http://www.Department0.University0.edu/FullProfessor200", "publicationAuthor"),
CI("http://www.Department1.University0.edu/FullProfessor210",
"Faculty"),
CI("http://www.Department12.University0.edu/UndergraduateStudent900",
"Student"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication210",
"http://www.Department0.University0.edu/FullProfessor3", "publicationAuthor"),
CI("http://www.Department14.University1.edu/GraduateStudent750",
"student"),
PI("http://www.Department14.University1.edu/GraduateStudent750",
"http://www.Department14.University1.edu/AssociateProfessor1", "advisor"),
CI("http://www.Department14.University1.edu/FullProfessor0/Publication610",
"Publication"),
PI("http://www.Department0.University0.edu/FullProfessor3",
"http://www.Department0.University0.edu/Course1", "teacherOf"),
PI("http://www.Department6.University0.edu/UndergraduateStudent54",
"http://www.Department6.University0.edu/Course1", "takesCourse"),
CI("http://www.Department0.University0.edu/FullProfessor123", "Faculty"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication1",
"http://www.Department0.University0.edu/FullProfessor100", "publicationAuthor"),
CI("http://www.Department0.University0.edu/FullProfessor123", "Faculty"),
CI("http://www.Department12.University0.edu/UndergraduateStudent888", "Student"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication80",
"http://www.Department0.University0.edu/FullProfessor2", "publicationAuthor"),
CI("http://www.Department14.University1.edu/GraduateStudent585", "student"),

```

```

PI("http://www.Department14.University1.edu/GraduateStudent585",
"http://www.Department14.University1.edu/AssociateProfessor1", "advisor"),
CI("http://www.Department14.University1.edu/FullProfessor0/Publication555",
"Publication"),
PI("http://www.Department0.University0.edu/FullProfessor12",
"http://www.Department0.University0.edu/Course1", "teacherOf"),
PI("http://www.Department6.University0.edu/UndergraduateStudent54",
"http://www.Department6.University0.edu/Course0", "takesCourse"),
CI("http://www.Department0.University0.edu/FullProfessor222", "Faculty"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication0",
"http://www.Department0.University0.edu/FullProfessor222", "publicationAuthor"),
CI("http://www.Department1.University0.edu/FullProfessor248", "Faculty"),
CI("http://www.Department12.University0.edu/UndergraduateStudent999", "Student"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication87",
"http://www.Department0.University0.edu/FullProfessor2", "publicationAuthor"),
CI("http://www.Department14.University1.edu/GraduateStudent880", "student"),
PI("http://www.Department14.University1.edu/GraduateStudent340",
"http://www.Department14.University1.edu/AssociateProfessor1", "advisor"),
CI("http://www.Department14.University1.edu/FullProfessor0/Publication890",
"Publication"),
PI("http://www.Department0.University0.edu/FullProfessor5",
"http://www.Department0.University0.edu/Course2", "teacherOf"),
PI("http://www.Department6.University0.edu/UndergraduateStudent77",
"http://www.Department6.University0.edu/Course1", "takesCourse")}

```

- 1 remoção: upd =

```

{NPI("http://www.Department0.University0.edu/FullProfessor0",
"http://www.Department0.University0.edu/GraduateCourse0", "teacherOf")}

```
- 5 remoções: upd =

```

{NCI("http://www.Department0.University0.edu/FullProfessor1",
"Faculty"),
NPI("http://www.Department0.University0.edu/FullProfessor0",
"http://www.Department0.University0.edu/GraduateCourse0", "teacherOf"),
NCI("http://www.Department0.University0.edu/FullProfessor0/Publication0",
"Publication"),
NCI("http://www.Department12.University0.edu/UndergraduateStudent350",
"Student"),
NPI("http://www.Department0.University0.edu/FullProfessor0/Publication10",
"http://www.Department0.University0.edu/FullProfessor0", "publicationAuthor")}

```
- 10 remoções: upd =

```

{NCI("http://www.Department0.University0.edu/FullProfessor1",
"Faculty"),
NPI("http://www.Department0.University0.edu/FullProfessor0",
"http://www.Department0.University0.edu/GraduateCourse0", "teacherOf"),
NCI("http://www.Department0.University0.edu/FullProfessor0/Publication0",
"Publication"),
NCI("http://www.Department12.University0.edu/UndergraduateStudent800",
"Student"),
NPI("http://www.Department0.University0.edu/FullProfessor0/Publication110",
"http://www.Department0.University0.edu/FullProfessor0", "publicationAuthor"),
NCI("http://www.Department14.University1.edu/GraduateStudent56",
"student"),
NPI("http://www.Department14.University1.edu/GraduateStudent650",
"http://www.Department14.University1.edu/AssociateProfessor4", "advisor"),
NCI("http://www.Department14.University1.edu/FullProfessor0/Publication510",
"Publication"),
NPI("http://www.Department0.University0.edu/FullProfessor9",

```

```
"http://www.Department0.University0.edu/Course14", "teacherOf"),
NPI("http://www.Department6.University0.edu/UndergraduateStudent298",
"http://www.Department6.University0.edu/Course0", "takesCourse")}
```

- 20 remoções: upd =


```
{NCI("http://www.Department0.University0.edu/FullProfessor1",
"Faculty"),
NPI("http://www.Department0.University0.edu/FullProfessor0",
"http://www.Department0.University0.edu/GraduateCourse0", "teacherOf"),
NCI("http://www.Department0.University0.edu/FullProfessor0/Publication0",
"Publication"),
NCI("http://www.Department12.University0.edu/UndergraduateStudent800",
"Student"),
NPI("http://www.Department0.University0.edu/FullProfessor0/Publication110",
"http://www.Department0.University0.edu/FullProfessor0", "publicationAuthor"),
NCI("http://www.Department14.University1.edu/GraduateStudent56",
"student"),
NPI("http://www.Department14.University1.edu/GraduateStudent650",
"http://www.Department14.University1.edu/AssociateProfessor4", "advisor"),
NCI("http://www.Department14.University1.edu/FullProfessor0/Publication510",
"Publication"),
NPI("http://www.Department0.University0.edu/FullProfessor9",
"http://www.Department0.University0.edu/Course14", "teacherOf"),
NPI("http://www.Department6.University0.edu/UndergraduateStudent298",
"http://www.Department6.University0.edu/Course0", "takesCourse"),
NCI("http://www.Department0.University0.edu/FullProfessor4",
"Faculty"),
NPI("http://www.Department0.University0.edu/FullProfessor1",
"http://www.Department0.University0.edu/GraduateCourse2", "teacherOf"),
NCI("http://www.Department0.University0.edu/FullProfessor0/Publication1",
"Publication"),
NCI("http://www.Department12.University0.edu/UndergraduateStudent20",
"Student"),
NPI("http://www.Department0.University0.edu/FullProfessor0/Publication1",
"http://www.Department0.University0.edu/FullProfessor0", "publicationAuthor"),
NCI("http://www.Department14.University1.edu/GraduateStudent3",
"Student"),
NPI("http://www.Department14.University1.edu/GraduateStudent5",
"http://www.Department14.University1.edu/FullProfessor6", "advisor"),
NCI("http://www.Department14.University1.edu/FullProfessor0/Publication8",
"Publication"),
NPI("http://www.Department9.University0.edu/FullProfessor0",
"http://www.Department9.University0.edu/Course0", "teacherOf"),
NPI("http://www.Department6.University0.edu/UndergraduateStudent34",
"http://www.Department6.University0.edu/Course3", "takesCourse")}
```
- 40 remoções: upd =


```
{NCI("http://www.Department0.University0.edu/FullProfessor1",
"Faculty"),
NPI("http://www.Department0.University0.edu/FullProfessor0",
"http://www.Department0.University0.edu/GraduateCourse0", "teacherOf"),
NCI("http://www.Department0.University0.edu/FullProfessor0/Publication0",
"Publication"),
NCI("http://www.Department12.University0.edu/UndergraduateStudent800",
"Student"),
NPI("http://www.Department0.University0.edu/FullProfessor0/Publication110",
"http://www.Department0.University0.edu/FullProfessor0", "publicationAuthor"),
NCI("http://www.Department14.University1.edu/GraduateStudent56",
"student"),
```

NPI("http://www.Department14.University1.edu/GraduateStudent650",
 "http://www.Department14.University1.edu/AssociateProfessor4", "advisor"),
 NCI("http://www.Department14.University1.edu/FullProfessor0/Publication510",
 "Publication"),
 NPI("http://www.Department0.University0.edu/FullProfessor9",
 "http://www.Department0.University0.edu/Course14", "teacherOf"),
 NPI("http://www.Department6.University0.edu/UndergraduateStudent298",
 "http://www.Department6.University0.edu/Course0", "takesCourse"),
 NCI("http://www.Department0.University0.edu/FullProfessor4",
 "Faculty"),
 NPI("http://www.Department0.University0.edu/FullProfessor1",
 "http://www.Department0.University0.edu/GraduateCourse2", "teacherOf"),
 NCI("http://www.Department0.University0.edu/FullProfessor0/Publication1",
 "Publication"),
 NCI("http://www.Department12.University0.edu/UndergraduateStudent20",
 "Student"),
 NPI("http://www.Department0.University0.edu/FullProfessor0/Publication1",
 "http://www.Department0.University0.edu/FullProfessor0", "publication.Author"),
 NCI("http://www.Department14.University1.edu/GraduateStudent3",
 "Student"),
 NPI("http://www.Department14.University1.edu/GraduateStudent5",
 "http://www.Department14.University1.edu/FullProfessor6", "advisor"),
 NCI("http://www.Department14.University1.edu/FullProfessor0/Publication8",
 "Publication"),
 NPI("http://www.Department9.University0.edu/FullProfessor0",
 "http://www.Department9.University0.edu/Course0", "teacherOf"),
 NPI("http://www.Department6.University0.edu/UndergraduateStudent34",
 "http://www.Department6.University0.edu/Course3", "takesCourse"),
 NCI("http://www.Department0.University1.edu/FullProfessor3",
 "Faculty"),
 NPI("http://www.Department1.University0.edu/FullProfessor0",
 "http://www.Department1.University0.edu/GraduateCourse0", "teacherOf"),
 NCI("http://www.Department0.University0.edu/FullProfessor0/Publication13",
 "Publication"),
 NCI("http://www.Department12.University0.edu/UndergraduateStudent66",
 "Student"),
 NPI("http://www.Department0.University0.edu/FullProfessor0/Publication8",
 "http://www.Department0.University0.edu/FullProfessor0", "publication.Author"),
 NCI("http://www.Department14.University1.edu/GraduateStudent33",
 "Student"),
 NPI("http://www.Department14.University1.edu/GraduateStudent10",
 "http://www.Department14.University1.edu/FullProfessor1", "advisor"),
 NCI("http://www.Department14.University1.edu/FullProfessor0/Publication11",
 "Publication"),
 NPI("http://www.Department3.University0.edu/FullProfessor7",
 "http://www.Department3.University0.edu/Course10", "teacherOf"),
 NPI("http://www.Department6.University0.edu/UndergraduateStudent10",
 "http://www.Department6.University0.edu/Course19", "takesCourse"),
 NCI("http://www.Department0.University0.edu/FullProfessor6",
 "Faculty"),
 NPI("http://www.Department3.University0.edu/AssociateProfessor8",
 "http://www.Department3.University0.edu/GraduateCourse28", "teacherOf"),
 NCI("http://www.Department0.University0.edu/FullProfessor5/Publication11",
 "Publication"),
 NCI("http://www.Department12.University0.edu/UndergraduateStudent44",
 "Student"),
 NPI("http://www.Department0.University1.edu/AssociateProfessor11/Publication4",
 "http://www.Department0.University1.edu/AssociateProfessor11", "publication.Author"),

```

NCI("http://www.Department14.University1.edu/GraduateStudent11",
"Student"),
NPI("http://www.Department14.University1.edu/GraduateStudent8",
"http://www.Department14.University1.edu/FullProfessor3", "advisor"),
NCI("http://www.Department14.University1.edu/FullProfessor0/Publication16",
"Publication"),
NPI("http://www.Department0.University1.edu/FullProfessor0",
"http://www.Department0.University1.edu/Course0", "teacherOf"),
NPI("http://www.Department6.University0.edu/UndergraduateStudent100",
"http://www.Department6.University0.edu/Course13", "takesCourse")}

```

- 7 inserções / 3 remoções: upd =

```

{CI("http://www.Department0.University0.edu/FullProfessor100",
"Faculty"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication0",
"http://www.Department0.University0.edu/FullProfessor100", "publicationAuthor"),
CI("http://www.Department0.University0.edu/FullProfessor110",
"Faculty"),
CI("http://www.Department12.University0.edu/UndergraduateStudent800",
"Student"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication110",
"http://www.Department0.University0.edu/FullProfessor0", "publicationAuthor"),
CI("http://www.Department14.University1.edu/FullProfessor0/Publication510",
"Publication"),
PI("http://www.Department0.University0.edu/FullProfessor12",
"http://www.Department0.University0.edu/Course0", "teacherOf"),
NCI("http://www.Department0.University0.edu/FullProfessor1",
"Faculty"),
NPI("http://www.Department0.University0.edu/FullProfessor0",
"http://www.Department0.University0.edu/GraduateCourse0", "teacherOf"),
NCI("http://www.Department0.University0.edu/FullProfessor0/Publication0",
"Publication")}

```
- 5 inserções / 5 remoções: upd =

```

{CI("http://www.Department14.University1.edu/FullProfessor0/Publication510",
"Publication"),
PI("http://www.Department0.University0.edu/FullProfessor12",
"http://www.Department0.University0.edu/Course0", "teacherOf"),
CI("http://www.Department0.University0.edu/FullProfessor110",
"Faculty"),
CI("http://www.Department12.University0.edu/UndergraduateStudent800",
"Student"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication110",
"http://www.Department0.University0.edu/FullProfessor0", "publicationAuthor"),
NCI("http://www.Department0.University0.edu/FullProfessor1",
"Faculty"),
NPI("http://www.Department0.University0.edu/FullProfessor0",
"http://www.Department0.University0.edu/GraduateCourse0", "teacherOf"),
NCI("http://www.Department0.University0.edu/FullProfessor0/Publication0",
"Publication"),
NCI("http://www.Department12.University0.edu/UndergraduateStudent800",
"Student"),
NPI("http://www.Department0.University0.edu/FullProfessor0/Publication110",
"http://www.Department0.University0.edu/FullProfessor0", "publicationAuthor")}

```
- 3 inserções / 7 remoções: upd =

```

{CI("http://www.Department0.University0.edu/FullProfessor110",
"Faculty"),
CI("http://www.Department12.University0.edu/UndergraduateStudent800",

```

```

"Student"),
PI("http://www.Department0.University0.edu/FullProfessor0/Publication110",
"http://www.Department0.University0.edu/FullProfessor0", "publicationAuthor"),
NCI("http://www.Department0.University0.edu/FullProfessor1",
"Faculty"),
NPI("http://www.Department0.University0.edu/FullProfessor0",
"http://www.Department0.University0.edu/GraduateCourse0", "teacherOf"),
NCI("http://www.Department0.University0.edu/FullProfessor0/Publication0",
"Publication"),
NCI("http://www.Department12.University0.edu/UndergraduateStudent800",
"Student"),
NPI("http://www.Department0.University0.edu/FullProfessor0/Publication110",
"http://www.Department0.University0.edu/FullProfessor0", "publicationAuthor"),
NPI("http://www.Department0.University0.edu/FullProfessor9",
"http://www.Department0.University0.edu/Course14", "teacherOf"),
NPI("http://www.Department6.University0.edu/UndergraduateStudent298",
"http://www.Department6.University0.edu/Course0", "takesCourse")}

```