Journal of Computer and System Sciences ••• (••••) •••–•••

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Expressive power of entity-linking frameworks ☆

Douglas Burdick [a], Ronald Fagin [a,*], Phokion G. Kolaitis [b,a], Lucian Popa [a], Wang-Chiew Tan [c]

[a] IBM Research – Almaden, United States of America
[b] University of California Santa Cruz, United States of America
[c] Megagon Labs, United States of America

ARTICLE INFO

Article history:
Received 12 October 2017
Received in revised form 22 June 2018
Accepted 10 September 2018
Available online xxxx

Keywords:
Entity-linking framework
Expressive power
Certain links

ABSTRACT

We develop a unifying approach to declarative entity linking by introducing the notion of an entity-linking framework and an accompanying notion of the certain links in such a framework. In an entity-linking framework, logic-based constraints are used to express properties of the desired link relations in terms of source relations and, possibly, in terms of other link relations. The definition of the certain links in such a framework makes use of weighted repairs and consistent answers in inconsistent databases. We demonstrate the modeling capabilities of this approach by showing that numerous concrete entity-linking scenarios can be cast as such entity-linking frameworks for suitable choices of constraints and weights. By using the certain links as a measure of expressive power, we investigate the relative expressive power of several entity-linking frameworks and obtain sharp comparisons.

1. Introduction and summary of results

Entity linking is the problem of creating links among records representing real-world entities that are related in certain ways. As an important special case, it includes *entity resolution*, which is the problem of identifying or linking "duplicate" entities. Since the pioneering work of Fellegi and Sunter [16] in 1969, entity linking has been recognized as a fundamental computational problem that has been investigated by several different research communities. While much of the work in this area [12,14,20,23] has focused and continues to focus on the design, implementation, and validation of direct algorithms for entity linking (and, in particular, for entity resolution), recent investigations have developed declarative approaches to entity linking that make it possible to separate the specification of entity linking from its actual implementation (see, for example, [1,9,18,19]).

In [9], we introduced and explored a declarative approach to entity linking that makes use of logical constraints. Our approach differs from earlier declarative approaches because it uses *link-to-source* constraints, instead of *source-to-link* constraints. Source-to-link constraints constitute, in effect, rules for creating links from source data in an operational manner. Approaches based on source-to-link constraints include Dedupalog [1] and HIL [19]. Another related approach is the one based on matching dependencies (MDs), introduced in [15], to enforce equality on attribute values based on matching con-

☆ This is the journal version of [10].

* Corresponding author.
E-mail address: fagin@us.ibm.com (R. Fagin).

https://doi.org/10.1016/j.jcss.2018.09.001
0022-0000/

ditions. In effect, MDs are source constraints that lead to in-place modifications of the source relations. MDs have been given operational semantics in [6] via a variation of the chase procedure that fixes violations of a given set of MDs.

Our link-to-source constraints spell out declarative conditions that the links must satisfy, independently of how the links will be created, and thus give rise to *solutions* of the declarative entity-linking specification at hand. In [9], we focused on the class of *maximum-value* solutions as "good" solutions for entity linking; intuitively, these are the solutions in which links have maximum "justification" in terms of the constraints and in terms of the source data. Since there can be multiple maximum-value solutions, we introduced the notion of the *certain links*, which, by definition, are the links that appear in every maximum-value solution and, therefore, are the links that should be kept. We then explored the problem of enumerating all maximum-value solutions and the problem of computing the certain links. This investigation was carried out for several different languages expressing link-to-source constraints, including languages that capture *collective entity resolution*, where interdependence between link relations is allowed. Unlike matching dependencies, all languages considered in [9] apply to scenarios in which the source instances are given as clean databases; therefore, the source instances should not be modified.

The variety and multitude of entity-linking approaches raise the question of developing methods and tools for comparing such different approaches. A comparative evaluation of the performance of several different direct algorithms for entity resolution (or entity matching) has been carried in [21] and [22]. Up to now, however, no methodology has been developed for comparing, along some axis, different declarative approaches for entity linking. The main aim of this article is to develop such a methodology that is centered on the notion of the *expressive power* of declarative entity-linking frameworks.

Our first conceptual contribution is to formulate a unifying notion of an *entity-linking framework* and an accompanying notion of the *certain links* in such a framework. This is achieved by bringing into the picture a notion of *weighted repairs* of inconsistent databases; these are a variant of the notion of weighted repairs of inconsistent databases in description logics studied in [13]. The "good" solutions for entity linking are then identified with the maximum weight repairs of inconsistent databases with respect to suitable choices of constraints and weights, while the certain links are defined to be the *consistent answers* of atomic link queries with respect to the maximum weight repairs, that is, those links that are in every maximum weight repair.

The inconsistent database whose weighted repairs we consider gives an upper bound or a domain for the candidate links; it could be provided (e.g., handed in from another system), or could be simply based on the Cartesian product of sets of entities (which we do in many of our definitions and proofs[1]).

This general approach gives rise to a single formalism for declarative entity linking in which the constraint language, the sets of constraints allowed, and the weight function that measures the "strength" of the links are parameters of the definition. We demonstrate the modeling capabilities of this formalism by showing, first, that it contains as special cases all but one of the concrete declarative entity-linking scenario studied in [9] (it does not capture the scenario of maximal solutions, a scenario that [9] says leads anyway to a "naive semantics"); further, our formalism accounts for new entity-linking scenarios, such as entity linking based on maximum cardinality repairs and entity linking with constraints that incorporate preferences.

Our second conceptual contribution is to use the certain links as a measure of the expressive power of an entity-linking framework and define what it means for an entity-linking framework to *subsume* another entity-linking framework. This makes it possible to compare different entity-linking frameworks along the axis of their expressive power.

As regards technical results, we first show that, under some mild hypotheses on entity-linking frameworks, it is possible to enumerate with polynomial delay all maximum weight repairs and to compute the certain links in polynomial time. This general result contains as special cases several similar results for concrete entity-linking scenarios obtained in [9]. Our main technical contribution, however, is to delineate the relative expressive power of different linking frameworks. Specifically, we show that the entity-linking framework of the maximum-value solutions considered in [9] and the entity-linking framework of maximum cardinality repairs introduced here are of incomparable expressive power, in the sense that neither of the two can subsume the other. We also show that the entity-linking framework for collective entity resolution where the constraints allow the link relations to depend on other link relations is strictly more expressive than in the case where constraints do not allow for interdependence among the link relations. This increase in expressive power takes place even when the dependencies among the link relations are non-recursive. Finally, we show that we also gain expressive power by adding preference constraints, which represent an additional, practical mechanism (see HIL [19]) for specifying the "good" links by letting a user explicitly, and declaratively, give priority to some types of links over other types of links. Concretely, we show that there is an entity-linking framework with preference constraints that is not subsumed by the entity-linking framework of maximum-value solutions (with no preference constraints).

In summary, the conceptual and technical contributions in this article provide a unifying approach to declarative entity linking and pave the way for the systematic comparative evaluation of different entity-linking frameworks.

This article is the full version of the conference paper [10]. The key difference between this article and [10] is that we include the proofs in this article. It should be pointed out that since the expressive power is measured via the certain links, proving that a specific entity linking framework is not subsumed by some other specific entity-linking framework is a much more challenging task than simply showing that the constraints defining the first framework are not logically equivalent to

[1] This is conceptual and it does not mean that the Cartesian product needs to be materialized.

Please cite this article in press as: D. Burdick et al., Expressive power of entity-linking frameworks, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.09.001

those defining the second framework. The proofs of our results about the expressive power of entity-linking frameworks involve a combination of special-purpose techniques with techniques from finite model theory. In particular, the proof of the result concerning the expressive power of entity-linking frameworks with preference constraints makes essential use of a locality theorem that is interesting in its own right.

2. Background on declarative entity linking

A *schema* $\mathbf{P}$ is a set of relation symbols, each with a designated arity. A $\mathbf{P}$-*instance* K is a collection of relations P^K interpreting the relation symbols P in $\mathbf{P}$. A *fact* of K is an expression $P^K(t)$ or, simply, $P(t)$, where P is a relation symbol in $\mathbf{P}$ and t is a tuple in the relation P^K.

We focus on declarative scenarios for entity linking, such as the ones considered in [9]. In such scenarios, $\mathbf{S}$ is the schema of *source* relations, while $\mathbf{L}$ is the schema of *link* relations, where each link relation is binary; we assume that $\mathbf{S}$ and $\mathbf{L}$ are disjoint. We use the notation $\mathbf{R} = \mathbf{S} \cup \mathbf{L}$ for the union of the two schemas. If I is an $\mathbf{S}$-instance and J is an $\mathbf{L}$-instance, then $\langle I, J\rangle$ denotes the $\mathbf{R}$-instance that is the union of I and J viewed as sets of facts.

Relation symbols in $\mathbf{S}$ will be referred to as *source* symbols, while relation symbols in $\mathbf{L}$ will be referred to as *link* symbols. Some source symbols may be interpreted by *built-in* relations, that is, such symbols may have the same interpretation on every allowable source instance. For example, a source symbol may stand for the substring relation between two strings, or it may stand for a user-defined predicate, such as similarity of names. If J is an $\mathbf{L}$-instance and $(a, b) \in L^J$ for some link symbol L in $\mathbf{L}$, then we say that (a, b) is a *link* of L in J. We sometimes write such a link as the fact $L^J(a, b)$, or $L(a, b)$ when J is clear from the context. We may also refer to J as a *link instance*.

An important feature of declarative entity linking is that the link relations are distinct from the source relations. The purpose of entity linking is to establish connections among source values (i.e., among various identifiers or names of entities) without changing any of the source data. A link relation is defined, implicitly, via link-to-source constraints that specify the properties that the link relation must satisfy with respect to the source data.

2.1. The language $\mathcal{L}_\mathbf{0}$ and entity-linking specifications based on $\mathcal{L}_\mathbf{0}$

We first revisit the language $\mathcal{L}_\mathbf{0}$ introduced in [9]; this language consists of three types of constraints.

- Inclusion dependencies of the form $L[X] \subseteq S[A]$ and $L[Y] \subseteq T[B]$, where L is a link symbol, and S and T are source symbols. We use X and Y to denote the first and the second attribute of L, while A and B denote attributes in relations S and T, respectively. Note that S and T could be the same source symbol.
- Functional dependencies (FDs) $L : X \to Y$ and $L : Y \to X$, where L is a link symbol and X and Y denote the attributes of L.
- Matching constraints of the form:

$$L(x, y) \to \forall \mathbf{u}(\psi(x, y, \mathbf{u}) \to \alpha_1 \vee \ldots \vee \alpha_k), \tag{1}$$

where L is a link symbol, $\psi(x, y, \mathbf{u})$ is a (possibly empty) conjunction of atomic formulas over $\mathbf{S}$ (with the requirement that the universally quantified variables $\mathbf{u}$ must occur in ψ), and where each α_i is of the form $\exists \mathbf{z_i}\, \phi_i(x, y, \mathbf{u}, \mathbf{z_i})$. Each ϕ_i is a conjunction of atomic formulas[2] over $\mathbf{S}$ or equalities. We assume that the variables in $\mathbf{z_i}$ are disjoint from the variables in ψ and from $\{x, y\}$. The variables x and y are universally quantified, but for simplicity of notation we omit their quantifiers. Note also that if ψ is empty, then formula (1) becomes

$$L(x, y) \to (\alpha_1 \vee \ldots \vee \alpha_k). \tag{2}$$

We will give shortly an example to illustrate the intuition behind these types of constraints. At a high-level, the motivation behind the use of disjunction in a matching constraint is that it lists all the possible matching conditions $\alpha_1, \ldots, \alpha_k$ for why a link $L(x, y)$ may exist (provided ψ holds). If a link $L(x, y)$ exists, then one or more of those conditions must be true (provided ψ holds). We do not require a matching constraint to be given for each link; for those links without a matching constraint, the link relation is restricted only by the rest of the constraints. The inclusion dependencies have the important role of specifying the domain of values that can be used to populate a link relation, while the functional dependencies encode basic cardinality constraints on the result of entity linking. (See also [19] for the significance of such constraints in practice.) The presence of one functional dependency means that the links are required to be many-to-one, that is, an entity on one side must be linked with at most one entity on the other side. The presence of both functional dependencies means that the links must be one-to-one.

While in general there could be more than two inclusion dependencies for each link, all the scenarios considered in [9] focused on the case of exactly two inclusion dependencies and exactly one matching constraint per link symbol. While

[2] Note that some of these atomic formulas may involve built-in relations.

Please cite this article in press as: D. Burdick et al., Expressive power of entity-linking frameworks, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.09.001

other combinations of constraints may also be meaningful (e.g., more than two inclusion dependencies per link, or more than one matching constraint per link), these restrictions have a good practical motivation, since they correspond to entity linking statements in the HIL language [19].

Definition 1. An *entity-linking specification* in $\mathcal{L}_\mathbf{0}$ is a triple $\mathcal{E} = (\mathbf{L}, \mathbf{S}, \Sigma)$,[3] where $\mathbf{L}$ is a link schema, $\mathbf{S}$ is a source schema, and Σ is a set of constraints containing, for each link symbol L in $\mathbf{L}$, at most one matching constraint in $\mathcal{L}_\mathbf{0}$, two inclusion dependencies (one for each attribute of L), and zero, one, or two functional dependencies.

Definition 2. Let $\mathcal{E} = (\mathbf{L}, \mathbf{S}, \Sigma)$ be an entity-linking specification, and let I be a source instance. A *solution* for I w.r.t. $\mathcal{E}$ is a link instance J such that the instance $\langle I, J\rangle$, obtained taking the union of the facts in I and J together, satisfies the constraints in Σ.

Example 1. [9] In this scenario, we link subsidiaries in one database with parent companies in another database. Consider the following source schema $\mathbf{S}$:

$$\texttt{Subsid}(sid, sname, location) \qquad \texttt{Company}(cid, cname, hdqrt)$$
$$\texttt{Exec}(eid, cid, name, title)$$

This source schema includes the relation symbols `Subsid` from the first database, and `Company` and `Exec` from the second database. The link schema $\mathbf{L}$ consists of a single link relation $L(sid, cid)$. A source instance I for $\mathbf{S}$ is given below as a set of facts:

$$\texttt{Subsid}(s_1, \text{“Citibank N.A.”}, \text{“New York”}) \qquad \texttt{Company}(c_1, \text{“Citigroup Inc”}, \text{“New York”})$$
$$\texttt{Subsid}(s_2, \text{“CIT Bank”}, \text{“Salt Lake City”}) \qquad \texttt{Company}(c_2, \text{“CIT Group Inc”}, \text{“New York”})$$
$$\texttt{Exec}(e_1, c_1, \text{“E. McQuade”}, \text{“CEO, Citibank N.A.”})$$

In the above, ‘Citigroup Inc” and “CIT Group Inc” are two different parent companies, and “Citibank N.A.” is the name of a true subsidiary of “Citigroup Inc”, while “CIT Bank” is the name of a true subsidiary of “CIT Group Inc”. The goal of entity linking is to identify links such as $L(s_1, c_1)$ and $L(s_2, c_2)$.

The following set Σ of constraints can be used to specify declaratively the properties of the link relation in terms of the source relations. First, Σ contains two inclusion dependencies $L[sid] \subseteq \texttt{Subsid}[sid]$, $L[cid] \subseteq \texttt{Company}[cid]$, and the functional dependency $L : sid \to cid$. While the inclusion dependencies specify where L is allowed to take values from, the functional dependency gives the additional requirement that the links must be many-to-one from *sid* to *cid* (i.e., every subsidiary must link to at most one parent company). Additionally, Σ includes the matching constraint:

$$\begin{aligned} L(sid, cid) \to \forall sn, loc, cn, hd\ (&\texttt{Subsid}(sid, sn, loc) \wedge \texttt{Company}(cid, cn, hd) \\ &\to (sn \sim cn) \\ &\vee \\ &\exists e, n, t\,(\texttt{Exec}(e, cid, n, t)\ \wedge \text{contains}(t, sn))\,), \end{aligned}$$

which lists all possible reasons as to why a link may exist. Concretely, if a subsidiary id (*sid*) and a company id (*cid*) are linked, then for every binding of `Subsid` and `Company` source tuples where *sid* and *cid* respectively occur, it must be that one of the two matching conditions holds: (1) there is a similarity in the names, as specified by $sn \sim cn$, or (2) there is some executive working for the company and this executive has a title that contains the subsidiary's name.

The following are solutions for I w.r.t. $\mathcal{E}$:

$$J_1 = \{L(s_1, c_1), L(s_2, c_1)\} \qquad J_2 = \{L(s_1, c_1), L(s_2, c_2)\}$$
$$J_3 = \{L(s_1, c_2), L(s_2, c_1)\} \qquad J_4 = \{L(s_1, c_2), L(s_2, c_2)\}$$

We assume here that the name similarity predicate $\sim$ evaluates to true for all pairs of subsidiary name and company name occurring in our instance I (thus, “Citibank N.A.” $\sim$ “Citigroup Inc” but also “Citibank N.A.” $\sim$ “CIT Group Inc”, and so on). Note that the link $L(s_1, c_1)$ satisfies both the $\sim$ predicate and the `Exec`-based condition, while other links satisfy only the $\sim$ predicate. Intuitively, the link $L(s_1, c_1)$ is a stronger link than the others. The link instance $J_5 = \{L(s_1, c_1), L(s_1, c_2)\}$ is not a solution, since it violates the functional dependency. Finally, note that a sub-instance of a solution is always a solution.

The above example illustrates that, in general, unlike the situation with entity resolution, in entity linking we allow linking of entities that are not necessarily of the same type; moreover, a link relation need not be an equivalence relation.

[3] [9] puts the link schema first to emphasize the link-to-source nature of the specification.

Please cite this article in press as: D. Burdick et al., Expressive power of entity-linking frameworks, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.09.001

2.2. Maximum-value solutions

An important feature of using link-to-source constraints for expressing matching requirements is that entity-linking specifications always have solutions (in particular, the empty solution). However, as it was observed in [9], not all solutions are equally good, for a given entity-linking specification and for a given source instance. As a result, [9] introduced a refinement on the notion of solutions, called "maximum-value solutions", based on assigning values to the links in a solution, where the value reflects intuitively the evidence supporting a link. We recall next the calculation of a value of a link from [9].

Given a set Σ of constraints in $\mathcal{L}_{\mathbf{0}}$ (with at most one matching constraint per link relation), an $\mathbf{R}$-instance $\langle I, J\rangle$ and a link fact $L^J(a,b)$, we define the value $\mathrm{Val}(L^J(a,b))$ of the link fact as follows. If $L(a,b)$ does not satisfy the inclusion dependencies, then $\mathrm{Val}(L^J(a,b)) = 0$. Otherwise, we distinguish several cases.

1. If Σ contains no matching constraint for L, then $\mathrm{Val}(L^J(a,b)) = 1$.
2. If Σ contains a matching constraint for L (which, by assumption, is the only such matching constraint) and if (a,b) does not satisfy the right-hand side of the matching constraint for L, then $\mathrm{Val}(L^J(a,b)) = 0$.
3. If Σ contains a matching constraint for L and if (a,b) satisfies the right-hand side of the matching constraint for L, we define $\mathrm{Val}(L^J(a,b))$ as follows.
   First, recall that the matching constraint for L has the form (1). Assume that there is no instantiation $\mathbf{u_0}$ of the vector of universally quantified variables $\mathbf{u}$ such that $I \models \psi(a,b,\mathbf{u}_0)$. This means that the matching constraint for $L(a,b)$ is satisfied for vacuous reasons. As in the earlier case of no matching constraint, we take the value of the link to be 1. In doing so, we treat the implication $\psi(x,y,\mathbf{u}) \rightarrow \alpha_1 \vee \ldots \vee \alpha_k$ in the matching constraint (1) as a strict implication rather than a material implication (i.e., we ignore the right-hand side when the left-hand side is false).
   In all other cases, we let the value of the link be:

$$\mathrm{Val}(L^J(a,b)) = \min_{\mathbf{u}_0}\Big(\sum_{\alpha_i,\mathbf{z}_0} 1\Big). \tag{3}$$

   In the above, $\mathbf{u}_0$ ranges over all the distinct instantiations of the vector of universally quantified variables $\mathbf{u}$ such that $I \models \psi(a,b,\mathbf{u}_0)$. We take the minimum, over all such $\mathbf{u}_0$, of the *strength* with which the source instance I satisfies the disjunction $\alpha_1 \vee \ldots \vee \alpha_k$. This strength is defined as a sum that gives a value of 1 for *every* distinct combination of a disjunct α_i such that I satisfies $\alpha_i(a,b,\mathbf{u}_0)$, and distinct instantiation $\mathbf{z}_0$ of the vector $\mathbf{z}$ of existentially quantified variables of α_i that makes the satisfaction of α_i hold. (Recall that α_i is, in general, of the form $\exists\mathbf{z}\,\phi_i(x,y,\mathbf{u},\mathbf{z})$.) In the case when α_i is satisfied and the existentially quantified variables are missing, then we count only 1. If ψ is empty, so that the matching constraint is of the form (2), then $\mathrm{Val}(L^J(a,b)) = \sum_{\alpha_i,\mathbf{z}_0} 1$.
   We can see that, intuitively, the sum in formula (3) calculates the strength of a link by counting the number of satisfied disjuncts together with the evidence (i.e., the number of existential witnesses). Taking the minimum guarantees that we take the weakest strength among all $\mathbf{u}_0$.

The value of a solution J, denoted by $\mathrm{Val}(J)$, is then the sum of the values of the links in J. Putting it all together, the following definition, adapted from [9], introduces the class of maximum-value solutions as well as the notion of certain links with respect to the class of maximum-value solutions.

Definition 3. Assume an entity-linking specification $\mathcal{E}$ in $\mathcal{L}_{\mathbf{0}}$. Given a source instance I, a *maximum-value solution* for I w.r.t. $\mathcal{E}$ is a link instance J such that: (1) J is a solution for I w.r.t. $\mathcal{E}$, and (2) for every other solution J', we have that $\mathrm{Val}(J') \leq \mathrm{Val}(J)$. The set of *certain links* for I w.r.t. the class of maximum-value solutions and $\mathcal{E}$ is the set of links that appear in every maximum-value solution J for I w.r.t. $\mathcal{E}$.

Let us revisit Example 1. We have that $\mathrm{Val}(L^{J_1}(s_1,c_1)) = 2$, since $L^{J_1}(s_1,c_1)$ satisfies both disjuncts in the matching constraint, while $\mathrm{Val}(L^{J_1}(s_2,c_1)) = 1$. Thus, the total value of the link instance J_1 is 3. Similarly, the other link instance containing $L(s_1,c_1)$, namely J_2, also has value 3. The remaining link instances J_3 and J_4 have value of 2. Hence, J_1 and J_2 are the two maximum-value solutions in this example. It follows that there is precisely one certain link, namely $L(s_1,c_1)$. This also reflects the intuition that $L(s_1,c_1)$ is the stronger link based on all the available evidence.

3. Entity-linking frameworks

In what follows, we will introduce the notion of an entity-linking framework, in which the constraint language, the sets of constraints allowed, and the weight function that measures the "strength" of the links are parameters of the framework.

3.1. Weighted repairs and consistent answers

We first consider a general setting where $\mathbf{S}$ and $\mathbf{L}$ are two disjoint relational schemas, and $\mathbf{R} = \mathbf{S} \cup \mathbf{L}$ is the union of these two schemas. In the subsequent subsection, we will instantiate this to the specific case where $\mathbf{S}$ is a source schema and $\mathbf{L}$ is the link schema.

Please cite this article in press as: D. Burdick et al., Expressive power of entity-linking frameworks, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.09.001

Definition 4. A *weight function on* **R** is a function w that assigns a non-negative weight $w(\langle I, J\rangle, L^J(a_1, \ldots, a_n))$ for every **R**-instance $\langle I, J\rangle$ and for every fact $L^J(a_1, \ldots, a_n)$ of J, where L is a relation symbol in **L**. The weight $w(\langle I, J\rangle, L^J(a_1, \ldots, a_n))$ is called the *weight* of the fact $L^J(a_1, \ldots, a_n)$ in $\langle I, J\rangle$.

Note that, even though only facts in relations interpreting **L**-symbols have weights, the weight of such a fact may depend on the entire **R**-instance $\langle I, J\rangle$ and not just on J.

In what follows, we will define the notion of a *maximum weight repair* of an **R**-instance $\langle I, J\rangle$; this notion is inspired by a similar one introduced by Du, Qi, and Shen [13] in the context of knowledge-bases with constraints expressed in description logics.

Definition 5. Let Σ be a set of integrity constraints on **R**, let w be a weight function on **R**, and let $\langle I, J\rangle$ be an **R**-instance. A sub-instance $\langle I, J'\rangle$ of $\langle I, J\rangle$ is a *maximum weight repair of* $\langle I, J\rangle$ *with respect to* Σ *and* w if $\langle I, J'\rangle$ has the following properties:

1. $\langle I, J'\rangle$ is consistent, i.e., $\langle I, J'\rangle$ satisfies every constraint in Σ.
2. J' has maximum weight, i.e., if $\langle I, J''\rangle$ is a consistent sub-instance of $\langle I, J\rangle$, then $\Sigma_{f\in J''} w(\langle I, J''\rangle, f) \leq \Sigma_{f\in J'} w(\langle I, J'\rangle, f)$.

In general, the weight function w may also depend on the set Σ of constraints at hand. If Σ and w are understood from the context, then we will simply talk about maximum weight repairs of $\langle I, J\rangle$, instead of maximum weight repairs of $\langle I, J\rangle$ with respect to Σ and w.

Thus, a maximum weight repair of $\langle I, J\rangle$ is a consistent sub-instance $\langle I, J'\rangle$ of $\langle I, J\rangle$ whose total sum of the weights of its **L**-facts is maximum across all consistent sub-instances $\langle I, J''\rangle$ of $\langle I, J\rangle$. In general, violations of integrity constraints can be repaired via tuple deletions, tuple insertions, or attribute value updates [11]. In this article, because of the constraints we allow, we consider only repairs obtained via tuple deletions from the link relations (source relations are not allowed to change).

Note that the notion of maximum weight repairs introduced in Definition 5 differs from the standard notion of subset repairs [2] in two ways: first, in the standard notion, the repair takes place with respect to the entire schema or, more precisely, we have there that $\mathbf{S} = \emptyset$ and $\mathbf{R} = \mathbf{L}$; second, in the standard notion, there is no weight function on the facts. Note also that *maximum cardinality subset repairs* [26] are the special case of maximum weight repairs in which $\mathbf{S} = \emptyset$, $\mathbf{R} = \mathbf{L}$, and the weight function assigns weight 1 to each fact. Finally, note that our notion of maximum weight repairs differs also from the notion of maximum weight repairs introduced in [13] in the following way. In [13], the weight of each fact f depends on the inconsistent instance $\langle I, J\rangle$ under consideration, but remains the same on all consistent sub-instances of $\langle I, J\rangle$ containing f. In contrast, in Definition 5, the weight of each fact f may differ from instance to instance; thus, we may have $w(\langle I, J\rangle, f) \neq w(\langle I, J'\rangle, f)$, where $\langle I, J'\rangle$ is a consistent sub-instance of $\langle I, J\rangle$.

Maximum weight repairs give rise to a notion of consistent answers of queries in exactly the same way subset repairs do.

Definition 6. Let Σ be a set of integrity constraints on **R** and let w be a weight function on **R**. If q is a query on **R**, and $\langle I, J\rangle$ is an **R**-instance, then a tuple **a** is a *consistent answer of* q *on* $\langle I, J\rangle$ *with respect to* Σ *and* w if $\mathbf{a} \in q(\langle I, J'\rangle)$, for every maximum weight repair $\langle I, J'\rangle$ of $\langle I, J\rangle$ with respect to Σ and w.

3.2. Certain links and entity-linking frameworks

We now consider the specific case of entity linking, where **S** is a source schema and **L** is the link schema.

Definition 7. Let **S** be a schema of source symbols, let **L** be a schema of link symbols, let Σ be a set of integrity constraints on $\mathbf{R} = \mathbf{S} \cup \mathbf{L}$, and let w be a weight function on $\mathbf{R} = \mathbf{S} \cup \mathbf{L}$. If L is a link symbol in **L** and $\langle I, J\rangle$ is an **R**-instance, then a *certain link of* L *on* $\langle I, J\rangle$ *with respect to* Σ *and* w is a consistent answer of the atomic query $L(x, y)$ on $\langle I, J\rangle$ with respect to Σ and w, i.e., a pair (a, b) such that $(a, b) \in L^{J'}$, for every maximum weight repair $\langle I, J'\rangle$ of $\langle I, J\rangle$ with respect to Σ and w.

We will also use the notation $L(a, b)$ for a certain link (a, b) of L. It will be clear from the context if $L(a, b)$ refers to a certain link or to a link $L^J(a, b)$ for some instance J.

Intuitively, in the above definition, we are given an instance $\langle I, J\rangle$, not necessarily consistent with respect to the set Σ of integrity constraints, where J represents an initial set of link facts. Then, the certain links of L on $\langle I, J\rangle$ represent precisely the subset of L-facts of J that appear in every maximum weight repair of $\langle I, J\rangle$. In this article, we focus on links that are certain, because this is a standard semantics in information integration, including data exchange and incomplete databases. While other alternatives may be considered (e.g., possible links, which are the links that appear in at least one maximum weight repair), we leave such investigation for future work. We point out, however, that the certain links have an advantage over the possible links because they provide a stronger guarantee.

Please cite this article in press as: D. Burdick et al., Expressive power of entity-linking frameworks, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.09.001

Note that Definition 7 is very general and does not make any assumptions about the class of integrity constraints that is allowed in Σ or about the weight function w. We also note that the weight function w is assumed to be defined over instances of $\mathbf{R} = \mathbf{S} \cup \mathbf{L}$, independently of whether these instances are consistent with Σ or not.

The concrete choices for Σ and w will be incorporated into the notion of *entity-linking frameworks*, which we define next, together with the notion of *entity-linking specifications*.

Definition 8. Let $\mathbf{S}$ be a schema of source symbols, let $\mathbf{L}$ be a schema of link symbols, and let $\mathbf{R} = \mathbf{S} \cup \mathbf{L}$.

- An *entity-linking framework on* $\mathbf{R}$ is a triple $(\mathcal{L}, \mathcal{S}, \mathcal{W})$ consisting of a logical language $\mathcal{L}$ on $\mathbf{R}$, a collection $\mathcal{S}$ of finite sets of $\mathcal{L}$-formulas, and a collection $\mathcal{W}$ of weight functions such that, for each $\Sigma \in \mathcal{S}$, there is a weight function w_Σ on $\mathbf{R}$.
- If Σ is a member of $\mathcal{S}$ and w_Σ is the associated weight function in $\mathcal{W}$, then we say that the triple $(\mathcal{L}, \Sigma, w_\Sigma)$ is an *entity-linking specification* in the entity-linking framework $(\mathcal{L}, \mathcal{S}, \mathcal{W})$.

Note that since $\mathcal{L}$ is a language on $\mathbf{R} = \mathbf{S} \cup \mathbf{L}$, Definition 8 allows for entity-linking frameworks that have source constraints, in addition to link constraints and constraints between links and sources. Here, we focus on entity-linking frameworks in which the language has no source constraints, because we assume that the source instances are given as clean databases that we do not need to modify.

Several different logical languages for expressing entity-linking specifications were introduced in [9] and then used to define and study different scenarios for declarative entity linking. Here, we show that all but one of the scenarios considered in [9] (namely, the scenario of maximal solutions) are concrete instances of the notion of an entity-linking framework in Definition 8, by choosing, in each case, the logical language $\mathcal{L}$, the collection $\mathcal{S}$ of finite sets of constraints from $\mathcal{L}$, and the collection $\mathcal{W}$ of weight functions. As we shall see, the weight functions can become progressively more sophisticated. Furthermore, the logical language $\mathcal{L}$ together with the collection $\mathcal{S}$ can become progressively richer.

3.3. Entity-linking frameworks based on $\mathcal{L}_\mathbf{0}$

As earlier mentioned, practical scenarios for entity linking using the language $\mathcal{L}_\mathbf{0}$ have focused on the case of exactly two inclusion dependencies and also on the case of exactly one matching constraint per link symbol [9]. The next definition captures these requirements by introducing the collection $\mathcal{S}_0$; it also introduces an initial instance $\langle I, I^* \rangle$ that will be used repeatedly in the sequel (intuitively, as a superset for the repairs).

Definition 9. Let $\mathbf{S}$ be a schema of source symbols and let $\mathbf{L}$ be a schema of link symbols.

- We write $\mathcal{S}_0$ to denote the collection of all finite sets Σ of $\mathcal{L}_\mathbf{0}$-formulas such that for each link symbol L, the set Σ contains one inclusion dependency on L for each of its attributes, contains zero, one or both functional dependencies on L, and at most one matching constraint on L.
- If I is an $\mathbf{S}$-instance, then we write I^* to denote the $\mathbf{L}$-instance defined as follows: for each link symbol L in $\mathbf{S}$, we have that $L^{I^*} = \pi_A(S^I) \times \pi_B(T^I)$, where A is the attribute of the source symbol S and B is the attribute of the source symbol T for which $\mathcal{L}_\mathbf{0}$ contains the inclusion dependencies $L[X] \subseteq S[A]$ and $L[Y] \subseteq T[B]$.

In the above definition, the instance $\langle I, I^* \rangle$ satisfies the inclusion dependencies of $\mathcal{L}_\mathbf{0}$ on each link symbol, but it need not satisfy the functional dependencies or the matching constraints of $\mathcal{L}_\mathbf{0}$.

We are now in a position to define several concrete entity-linking frameworks by instantiating the general concepts introduced above. We consider three different entity-linking frameworks obtained from $\mathcal{L}_\mathbf{0}$ and $\mathcal{S}_0$ by using three different types of weight functions.

Framework 1. The entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ of maximum-value solutions.

For each $\Sigma \in \mathcal{S}_0$, consider the following weight function w_Σ. Given an $\mathbf{R}$-instance $\langle I, J \rangle$ and a fact $L^J(a, b)$, we define $w_\Sigma(\langle I, J \rangle, L^J(a, b)) = \mathrm{Val}(L^J(a, b))$, where $\mathrm{Val}(L^J(a, b))$ is defined in Section 2 (relative to the set Σ of constraints).

By analyzing the definition of $\mathrm{Val}(L^J(a, b))$ we can see that the weight $w_\Sigma(\langle I, J \rangle, L^J(a, b))$ used by the above entity-linking framework does not actually depend on the link instance J but rather on the link fact itself.

Consider the above entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$. It is easy to verify that if I is an $\mathbf{S}$-instance, then the following statements are equivalent for an $\mathbf{L}$-instance J:

1. $\langle I, J \rangle$ is a maximum weight repair of $\langle I, I^* \rangle$ with respect to Σ and w_Σ.
2. J is a maximum-value solution for I with respect to Σ, as defined in [9].

It follows that the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ coincides with the entity-linking scenario given by $\mathcal{L}_\mathbf{0}(\oplus)$ in [9].

Please cite this article in press as: D. Burdick et al., Expressive power of entity-linking frameworks, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.09.001

Framework 2. The entity-linking frameworks $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{w}})$ of maximum-value solutions with weighted disjuncts.

For each matching constraint $L(x, y) \rightarrow \forall \mathbf{u}(\psi(x, y, \mathbf{u}) \rightarrow \alpha_1 \vee \ldots \vee \alpha_k)$ of $\mathcal{L}_{\mathbf{0}}$ and for each disjunct $\alpha_i ::= \exists \mathbf{z}\, \phi_i(x, y, \mathbf{u}, \mathbf{z})$, let $w_{\phi_i}(x, y, \mathbf{u}, \mathbf{z})$ be a function that returns non-negative numbers. Intuitively, with each disjunct that returns true or false, we also have a function that computes a weight for that disjunct. This collection of functions w_{ϕ_i} gives rise to a weight function $\mathcal{V}_{\mathbf{w}}$ that is computed as in the case of $\mathcal{V}_0$ except that in formula (3) we replace the number 1 by $w_{\phi_i}(a, b, \mathbf{u}_0, \mathbf{z}_0)$.

Note that each different collection of functions w_{ϕ_i} gives rise to a different entity-linking framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{w}})$. This family of frameworks captures the entity-linking scenarios given by $\mathcal{L}_{\mathbf{0}}(\oplus, \mathbf{w})$, which, as discussed in [9], is of special interest because of its connection to probabilistic methods for entity resolution, including those based on Markov Logic Networks (MLNs) [27].

Framework 3. The entity-linking framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{W}_{\mathbf{1}})$ of maximum cardinality repairs.

Let $\mathbf{1}$ be the weight function on $\mathbf{R}$ such that $\mathbf{1}(\langle I, J\rangle, L^J(a, b)) = 1$, for every $\mathbf{R}$-instance $\langle I, J\rangle$ and every fact $L^J(a, b)$. Consider the entity-linking framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{W}_{\mathbf{1}})$, where, for each $\Sigma \in \mathcal{S}_0$, we have that $w_\Sigma = \mathbf{1}$.

A maximum weight repair of $\langle I, I^*\rangle$ with respect to Σ and $\mathbf{1}$ is a repair that maximizes the total cardinality of the link facts. We call such repairs *maximum cardinality repairs*.

This is a new framework that has not been considered in [9]. It can be verified that if $\langle I, J\rangle$ is such a maximum cardinality repair of $\langle I, I^*\rangle$, then J is a maximal solution for I, as defined in [9]. The converse, however, does not always hold. Like maximal solutions, the notion of maximum cardinality repairs suffers from the deficiency that they give rise to "too few" certain links. This can be seen in the following example from [9].

Example 2. Assume the same schemas and constraints as in Example 1. Also, assume the same source instance I. It can be seen that, given our set Σ of constraints, there are exactly four maximum cardinality repairs for $\langle I, I^*\rangle$, namely $\langle I, J_i\rangle$, $i = 1, 4$, where the J_i's are the same as the solutions listed in Example 1. We display them again for convenience:

$$J_1 = \{L(s_1, c_1), L(s_2, c_1)\} \qquad J_2 = \{L(s_1, c_1), L(s_2, c_2)\}$$
$$J_3 = \{L(s_1, c_2), L(s_2, c_1)\} \qquad J_4 = \{L(s_1, c_2), L(s_2, c_2)\}$$

It follows that the set of certain links of L on $\langle I, I^*\rangle$ w.r.t. Σ and $\mathbf{1}$ is empty: there is no link that appears in all four maximum cardinality repairs and, hence, no link qualifies as a certain link. This is in contrast with the framework of maximum-value solutions, which is able, for the same example, to differentiate the link $L(s_1, c_1)$ as a stronger link than the other links. Recall from Section 2.2 that J_1 and J_2 are the maximum-value solutions for this example; hence, $L(s_1, c_1)$ is the certain link with respect to maximum-value solutions. However, the constant weight function $\mathbf{1}$ used by the simpler framework of maximum cardinality repairs does not provide such differentiation.

Next, we state a general theorem for enumerating all maximum weight repairs with polynomial delay and for computing the certain links in polynomial time. Several results in [9], including Theorem 5.4, are special cases of this theorem.

Theorem 1. *Let $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{W})$ be an entity-linking framework such that for each $\Sigma \in \mathbf{S}_0$, for each $\mathbf{S}$-instance I, for each sub-instance J of I^*, and for each fact $L^J(a, b)$, we have that $w_\Sigma(\langle I, I^*\rangle, L^{I^*}(a, b)) = w_\Sigma(\langle I, J\rangle, L^J(a, b))$. Then the following statements are true.*

1. *There is a polynomial-delay algorithm that, given an $\mathbf{S}$-instance I, enumerates the maximum weight repairs of $\langle I, I^*\rangle$.*
2. *There is a polynomial-time algorithm that, given an $\mathbf{S}$-instance I, computes the certain links of $\langle I, I^*\rangle$ with respect to Σ and w_Σ.*

Note that the hypothesis of Theorem 1 is satisfied by the preceding three entity-linking frameworks. In particular, in all three frameworks, the weight of a link fact does not depend on the link instance J in which it appears. The proof of Theorem 1 is essentially the same as the proof of Theorem 5.4 in [9], where the problem is reduced to computing and enumerating maximum-weight matchings in undirected weighted bipartite graphs.

3.4. Collective entity-linking frameworks

We now consider a language $\mathcal{L}_{\mathbf{c}}$ that is richer than $\mathcal{L}_{\mathbf{0}}$ and allows for link relations to appear in the right-hand side of matching constraints. Thus, the language $\mathcal{L}_{\mathbf{c}}$ allows us to express what is usually called *collective entity linking* [7], that is, the process of creating or specifying multiple inter-dependent links.

Concretely, in $\mathcal{L}_{\mathbf{c}}$, the matching constraint for a link symbol L has the same form

$$L(x, y) \rightarrow \forall \mathbf{u}(\psi(x, y, \mathbf{u}) \rightarrow \alpha_1 \vee \ldots \vee \alpha_k)$$

as in $\mathcal{L}_{\mathbf{0}}$, with the difference that in each disjunct $\alpha_i ::= \exists \mathbf{z}\, \phi_i(x, y, \mathbf{u}, \mathbf{z})$, the formula ϕ_i can now be a conjunction of source *and link* atomic formulas, along with equalities. Thus, the matching constraint for L is allowed to refer to other link symbols (possibly, including L itself). As an example, which we give shortly, in $\mathcal{L}_{\mathbf{c}}$ one can express matching constraints to

specify both publication links and venue links, where the matching constraint for publication links may depend on the links between venues, and the matching constraint for venue links may depend on the links between publications.

Based on the language $\mathcal{L}_{\mathbf{c}}$, we can define two entity-linking frameworks, one that does not allow for recursion among the links, and one that does allow for recursion.

Framework 4. The entity-linking framework $(\mathcal{L}_{\mathbf{c}}, \mathcal{S}_{\mathbf{1}}, \mathcal{V}_{\mathbf{1}})$ for recursion-free collective entity linking.

In this framework, $\mathcal{S}_1$ is the collection of all finite sets of constraints from $\mathcal{L}_{\mathbf{c}}$, such that for each link symbol L, the set Σ contains the two inclusion dependencies on L, it contains zero, one or two functional dependencies on L, and at most one matching constraint on L. Additionally, we require that there is no recursion through the links. Thus, for each Σ in $\mathcal{S}_1$, there is implicitly a hierarchy of link symbols, and a matching constraint for L may call only links that are strictly lower in the hierarchy than L. Additionally, $\mathcal{V}_1$ is the collection of weight functions that associates with each Σ in $\mathcal{S}_1$ a weight function w_Σ defined in the same way as in the entity-linking framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$.

Framework 5. The entity-linking framework $(\mathcal{L}_{\mathbf{c}}, \mathcal{S}_{\mathbf{2}}, \mathcal{V}_{\mathbf{2}})$ for recursive collective entity linking is defined in the same way as $(\mathcal{L}_{\mathbf{c}}, \mathcal{S}_{\mathbf{1}}, \mathcal{V}_{\mathbf{1}})$ except that $\mathcal{S}_2$ allows recursion through the links.

Example 3. Consider a bibliographic example from [9], where we link papers from one database with articles from another database, while also linking the corresponding venues. Other examples involving bibliographical data have appeared in the literature, including [5,7]. In our example, the source schema **S** consists of $\texttt{Paper}(pid, title, venue, year)$ and $\texttt{Article}(ano, title, journal, year)$. Here, `pid` is a unique id assigned to `Paper` records, while `venue` could be a conference, a journal, or some other place of publication. The `Article` relation represents publications that appeared in journals, and `ano` is a unique id assigned to such records. The link schema **L** consists of two relations: $\texttt{PaperLink}\ (pid, ano)$ and $\texttt{VenueLink}\ (venue, journal)$. The first relation is intended to link paper ids from `Paper` with article numbers from `Article`, when they represent the same publication. The second relation is intended to relate `journal` values that occur in `Article` (e.g., "ACM TODS") to journal values that occur under the `venue` field in `Paper` (e.g., "TODS").

A possible entity-linking specification in the framework $(\mathcal{L}_{\mathbf{c}}, \mathcal{S}_{\mathbf{2}}, \mathcal{V}_{\mathbf{2}})$ is $(\mathcal{L}_{\mathbf{c}}, \Sigma, w_\Sigma)$, where Σ contains the following two matching constraints:

$$\begin{aligned}
&\texttt{VenueLink}(ven, jou) \rightarrow (ven \sim_1 jou) \\
&\qquad \vee \exists pid, t_1, y_1, ano, t_2, y_2\ (\ \texttt{Paper}(pid, t_1, ven, y_1) \\
&\qquad\qquad \wedge\ \texttt{Article}(ano, t_2, jou, y_2) \\
&\qquad\qquad \wedge\ \texttt{PaperLink}(pid, ano)\ )
\end{aligned}$$

$$\begin{aligned}
&\texttt{PaperLink}(pid, ano) \rightarrow \\
&\quad \forall t_1, ven, y_1, t_2, jou, y_2\ (\ \texttt{Paper}(pid, t_1, ven, y_1) \wedge \texttt{Article}(ano, t_2, jou, y_2) \\
&\qquad\qquad \rightarrow ((t_1 \sim_2 t_2) \wedge (y_1 = y_2)) \\
&\qquad\qquad \vee ((t_1 \sim_2 t_2) \wedge \texttt{VenueLink}(ven, jou))\ )
\end{aligned}$$

The first constraint specifies that we may link a venue with a journal only if their string values are similar (via some similarity predicate $\sim_1$), or if there are papers and articles that have been published in the respective venue and journal and that are linked via `PaperLink`. The second constraint specifies that we may link a paper with an article only if their titles are similar (via a similarity predicate $\sim_2$) and their years of publication match exactly, or if their titles are similar and their venues of publications are linked via `VenueLink`.

Additionally, Σ includes two functional dependencies on `PaperLink`: $pid \rightarrow ano$, $ano \rightarrow pid$, to reflect that each paper id in `Paper` must match to at most one article number in `Article`, and vice-versa. We do not require any functional dependencies on `VenueLink`; thus, we could have multiple venue strings in `Paper` matching with a journal string in `Article`, and vice-versa. We also include in Σ the expected inclusion dependencies from the link attributes to the corresponding source attributes (e.g., $\texttt{PaperLink}[pid] \subseteq \texttt{Paper}[pid]$).

With a simple modification, where we remove the second disjunct in the matching constraint for `PaperLink`, we obtain a different entity-linking specification that is in the recursion-free collective entity-linking framework $(\mathcal{L}_{\mathbf{c}}, \mathcal{S}_{\mathbf{1}}, \mathcal{V}_{\mathbf{1}})$.

We point out that the entity-linking framework $(\mathcal{L}_{\mathbf{c}}, \mathcal{S}_{\mathbf{1}}, \mathcal{V}_{\mathbf{1}})$ coincides with the entity-linking scenario given by $\mathcal{L}_{\mathbf{1}}(\oplus)$ in [9], while entity-linking framework $(\mathcal{L}_{\mathbf{c}}, \mathcal{S}_{\mathbf{2}}, \mathcal{V}_{\mathbf{2}})$ coincides with the entity-linking scenario given by $\mathcal{L}_{\mathbf{2}}(\oplus)$ in [9]. This is yet another manifestation of the modeling capabilities of the general notion of an entity-linking framework in Definition 8.

For the preceding two entity-linking frameworks (Framework 4 and Framework 5), it is important to note that the weight functions depend on the link instance in a crucial way. In particular, the hypothesis of the preceding Theorem 1, stating that the weight of a link fact only depends on I^* and not on the link instance J, is no longer satisfied. In fact, as shown in [9] (Theorem 7.3), Theorem 1 fails even for $(\mathcal{L}_{\mathbf{c}}, \mathcal{S}_{\mathbf{1}}, \mathcal{V}_{\mathbf{1}})$, unless NP = coNP.

Please cite this article in press as: D. Burdick et al., Expressive power of entity-linking frameworks, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.09.001

4. Comparing the expressive power of entity-linking frameworks

The notion of certain links makes it possible to compare the expressive power of entity-linking frameworks. In the next definition, we first introduce the notion of *certain-link equivalence* between entity-linking specifications. This notion is of interest as a tool to compare entity-linking specifications in a way other than logical equivalence (which may be too strict for entity linking purposes). The second part of the definition then makes use of certain-link equivalence to define a notion of *subsumption* between entity-linking frameworks.

Definition 10. Let $\mathbf{S}$ be a schema of source symbols, let $\mathbf{L}$ be a schema of link symbols, let $\mathbf{R} = \mathbf{S} \cup \mathbf{L}$. Assume that $\mathcal{F} = (\mathcal{L}, \mathcal{S}, \mathcal{W})$ and $\mathcal{F}' = (\mathcal{L}', \mathcal{S}', \mathcal{W}')$ are two entity-linking frameworks on $\mathbf{R}$.

- Let $\mathcal{E} = (\mathcal{L}, \Sigma, w_\Sigma)$ be an entity-linking specification in $\mathcal{F}$, and let $\mathcal{E}' = (\mathcal{L}', \Sigma', w_{\Sigma'})$ be an entity-linking specification in $\mathcal{F}'$. We say that $\mathcal{E}$ *and* $\mathcal{E}'$ *are certain-link equivalent* if for every link symbol L in $\mathbf{L}$ and every $\mathbf{R}$-instance $\langle I, J\rangle$, we have that the certain links of L on $\langle I, J\rangle$ with respect to Σ and w_Σ coincide with the certain links of L on $\langle I, J\rangle$ with respect to Σ' and $w_{\Sigma'}$.
- We say that $\mathcal{F}$ *is subsumed by* $\mathcal{F}'$, denoted $\mathcal{F} \preceq \mathcal{F}'$, if for every entity-linking specification $\mathcal{E}$ of $\mathcal{F}$ there is an entity-linking specification $\mathcal{E}'$ of $\mathcal{F}'$ such that $\mathcal{E}$ and $\mathcal{E}'$ are certain-link equivalent. Otherwise, we say that $\mathcal{F}$ *is not subsumed by* $\mathcal{F}'$, and write $\mathcal{F} \not\preceq \mathcal{F}'$.
- We say that $\mathcal{F}$ *is strictly subsumed by* $\mathcal{F}'$ if $\mathcal{F} \preceq \mathcal{F}'$, but $\mathcal{F}' \not\preceq \mathcal{F}$.

We note that a weaker notion of subsumption was considered implicitly in [9] for concrete entity-linking scenarios. In this weaker notion, we say that $\mathcal{E} = (\mathcal{L}, \Sigma, w_\Sigma)$ and $\mathcal{E}' = (\mathcal{L}', \Sigma', w_{\Sigma'})$ are certain-link equivalent if for every link symbol L in $\mathbf{L}$ we have that the certain links of L on $\langle I, I^*\rangle$ with respect to Σ and w_Σ coincide with the certain links of L on $\langle I, I^*\rangle$ with respect to Σ' and $w_{\Sigma'}$. Thus, this weaker notion considers only repairs of the instance $\langle I, I^*\rangle$ instead of repairs of arbitrary instances $\langle I, J\rangle$.

We note that for of all our subsumption results (Theorems 2, 3, 4, and 5), whenever we prove failure of subsumption, we actually prove it in a stronger sense, by showing that it fails even under the weaker notion.

The next two theorems say that the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ of maximum-value solutions and the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{W}_\mathbf{1})$ of maximum cardinality repairs are incomparable in expressive power, in that neither subsumes the other.

Theorem 2. *The entity-linking framework* $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ *of maximum-value solutions is not subsumed by the entity-linking framework* $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{W}_\mathbf{1})$ *of maximum cardinality repairs.*

Proof. We now give an entity-linking specification $\mathcal{E}$ in $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ for which there is no $\mathcal{E}'$ in $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{W}_\mathbf{1})$ that is certain-link equivalent to it.

Let $\mathcal{E}$ be the entity-linking specification given by the following:

- $L(x, y) \to R(x, y) \vee S(x, y) \vee T(x, y)$
- FD $L: X \to Y$
- $L[X] \subseteq D$
- $L[Y] \subseteq D$

Note that the relation D, which appears in the inclusion dependencies, is unary.

For each of the instances I_j we now specify, the role of J in Definition 10 is played by $(I_j)^*$, which is defined in Definition 9.

For $I_1 = \{R(0,1), R(0,2), S(0,2), D(0), D(1), D(2)\}$, we have $L(0,2)$ as a certain link.

For $I_2 = \{R(0,1), R(0,2), S(0,1), S(0,2), T(0,1), D(0), D(1), D(2)\}$, we have $L(0,1)$ as a certain link.

For $I_3 = \{R(0,1), R(0,2), T(0,2), D(0), D(1), D(2)\}$, we have $L(0,2)$ as a certain link.

For $I_4 = \{R(0,1), R(1,1), D(0), D(1)\}$, we have $L(0,1)$ and $L(1,1)$ as certain links.

For $I_5 = \{R(0,1), S(1,2), D(0), D(1), D(2)\}$, we have $L(0,1)$ and $L(1,2)$ as certain links.

Assume that there were an entity-linking specification $\mathcal{E}'$ in the framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{W}_\mathbf{1})$ of maximum cardinality repairs that is certain-link equivalent to $\mathcal{E}$; we shall derive a contradiction.

Because of I_5, we see that $\mathcal{E}'$ has the inclusion dependencies $L[X] \subseteq D$ and $L[Y] \subseteq D$, since no projection of R, S, or T contains all of the values of $L[X]$ (respectively, of $L[Y]$).

Because of I_4, we see that $\mathcal{E}'$ does not have the FD $L: Y \to X$. So it either has no FDs or only the FD $L: X \to Y$.

If $\mathcal{E}'$ has no matching constraint, then $L(0,1)$ and $L(0,2)$ cannot be distinguished from each other in I_1, so either neither or both would be certain links, a contradiction.

Let the matching constraint for $\mathcal{E}'$ be of the form (1), with the same restrictions as given there. The way that we shall derive a contradiction is to show that $L(0,2)$ satisfies the marching constraint in I_2, that is, that the right-hand side of

(1) holds when $x = 0$ and $y = 2$. Thus, assume that $L(0, 2)$ satisfies the marching constraint in I_2. Let M be a maximum cardinality repair for I_2. Then M contains $L(0, 1)$, since $L(0, 1)$ is a certain link for I_2. Form M' from M by replacing $L(0, 1)$ by $L(0, 2)$. Then M' satisfies the inclusion dependencies and the only possible FD $L: X \to Y$. Also, by assumption, $L(0, 2)$ satisfies the matching constraint. Hence, M' is a maximum cardinality repair. But this is a contradiction, since M' does not contain the certain link $L(0, 1)$. So the proof is complete if we show that $L(0, 2)$ satisfies the marching constraint in I_2.

We begin by considering the case where ψ is empty. Thus, in this case the matching constraint is of the form (2) in Section 2.

Then some disjunct α_i is satisfied in I_1 when $x = 0$ and $y = 2$. But that same disjunct is satisfied in I_2 when $x = 0$ and $y = 2$, since I_1 is a sub-instance of I_2. So $L(0, 2)$ satisfies the matching constraint in I_2, as desired.

Now consider the other case, where ψ is nonempty. If ψ were to contain S, then the matching constraint (1) would be trivially satisfied in I_3 when $x = 0$ and $y = 1$, since there is no atomic fact involving S in I_3. Let M be a maximum cardinality repair for I_3. Then M contains $L(0, 2)$, since $L(0, 2)$ is a certain link for I_3. Form M' from M by replacing $L(0, 2)$ by $L(0, 1)$. Then M' satisfies the inclusion dependencies and the only possible FD $L: X \to Y$. And as we already noted, $L(0, 1)$ satisfies the matching constraint. Hence, M' is a maximum cardinality repair. But this is a contradiction, since M' does not contain the certain link $L(0, 2)$. So ψ does not contain S. Similarly, by considering I_1, we see that ψ does not contain T. Since ψ does not contain S or T, it follows that ψ contains only R and/or D.

Let $x = 0$ and $y = 2$. We now show that

$$\forall \mathbf{u}(\psi(x, y, \mathbf{u}) \to \alpha_1 \vee \ldots \vee \alpha_k) \tag{4}$$

holds in I_2. This certainly holds if there is no assignment t to the variables in $\mathbf{u}$ that makes $\psi(x, y, \mathbf{u})$ hold in I_2 when $x = 0$ and $y = 2$. So let t be an arbitrary assignment to the variables in $\mathbf{u}$ that makes $\psi(x, y, \mathbf{u})$ hold in I_2 when $x = 0$ and $y = 2$, and let t' be the extension of t where $t'(x) = 0$ and $t'(y) = 2$. Then ψ also holds in I_1 under t', since I_1 and I_2 agree on the tuples of R and D, the only possible relation symbols of ψ. Then there is some disjunct α_i that holds in I_1 under t', since $L(0, 2)$ is a certain link for I_1. Now α_i is of the form $\exists \mathbf{z_i}\, \phi_i(x, y, \mathbf{u}, \mathbf{z_i})$, where ϕ_i is a conjunction of atomic formulas and equalities. Since α_i holds in I_1 under t', there is t'' that extends t' to the variables in $\mathbf{z_i}$ such that ϕ_i holds in I_1 under t''. Then ϕ_i holds in I_2 under t'', since (a) every atomic fact that holds in I_1 also holds in I_2, and (b) the equalities among variables in t'' hold independent of which database we are considering. So $\psi(x, y, \mathbf{u}) \to \alpha_i$ holds in I_2 under t when $x = 0$ and $y = 2$, and hence

$$\psi(x, y, \mathbf{u}) \to \alpha_1 \vee \ldots \vee \alpha_k$$

holds in I_2 under t when $x = 0$ and $y = 2$. Since t is an arbitrary assignment to the variables in $\mathbf{u}$ that makes ψ hold in I_2 when $x = 0$ and $y = 2$, it follows that (4) holds in I_2 when $x = 0$ and $y = 2$. So $L(0, 2)$ satisfies the matching constraint in I_2, as desired. □

Theorem 3. *The entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{W}_\mathbf{1})$ of maximum cardinality repairs is not subsumed by the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ of maximum-value solutions.*

The proof of Theorem 3 is somewhat long and technical, and is given in the appendix.

By definition, the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ is subsumed by the entity-linking framework $(\mathcal{L}_\mathbf{c}, \mathcal{S}_\mathbf{1}, \mathcal{V}_\mathbf{1})$. The next theorem says that this subsumption is strict. This means that allowing for link relations to appear on the right-hand side of matching constraints gives strictly more expressive power than not allowing this, even when the dependencies among the link relations are non-recursive.

Theorem 4. *The entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ of maximum-value solutions is strictly subsumed by the entity-linking framework $(\mathcal{L}_\mathbf{c}, \mathcal{S}_\mathbf{1}, \mathcal{V}_\mathbf{1})$ for recursion-free collective entity linking.*

Proof. By definition, the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ is subsumed by the entity-linking framework $(\mathcal{L}_\mathbf{c}, \mathcal{S}_\mathbf{1}, \mathcal{V}_\mathbf{1})$. We now show that this subsumption is strict. Thus, we now give an entity-linking specification $\mathcal{E}$ in $(\mathcal{L}_\mathbf{c}, \mathcal{S}_\mathbf{1}, \mathcal{V}_\mathbf{1})$ for which there is no $\mathcal{E}'$ in $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ that is certain-link equivalent to it.

Let $\mathcal{E}$ be the entity-linking specification given by the following:

- $L_1(x, y) \to (S(x, y) \to (L_2(x, y) \wedge R(x, y))$
- $L_2(x, y) \to (P(x, y) \to T(x, y))$
- There are no FDs
- $L_1[X] \subseteq D$
- $L_1[Y] \subseteq D$
- $L_2[X] \subseteq D$
- $L_2[Y] \subseteq D$

Please cite this article in press as: D. Burdick et al., Expressive power of entity-linking frameworks, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.09.001

Note that the relation D, which appears in the inclusion dependencies, is unary.

For each of the instances I_j we now specify, the role of J in Definition 10 is played by $(I_j)^*$, which is defined in Definition 9.

For $I_1 = \{D(0)\}$, the link $L_1(0,0)$ is a certain link.

For $I_2 = \{D(0), S(0,0)\}$, the link $L_1(0,0)$ is not a certain link. This is because $R(0,0)$ does not hold, and so the matching constraint for $L_(0,0)$ is not satisfied.

For $I_3 = \{D(0), S(0,0), R(0,0)\}$, the link $L_1(0,0)$ is a certain link.

For $I_4 = \{D(0), S(0,0), R(0,0), P(0,0)\}$, the link $L_1(0,0)$ is not a certain link, since the matching constraint for $L_1(0,0)$ fails, as we now show. The matching constraint for $L_2(0,0)$ is not satisfied (since $P(0,0)$ holds but not $T(0,0)$). Since $L_2(0,0)$ fails, while $S(0,0)$ holds, we see that $L_1(0,0)$ does not satisfy the matching constraint for L_1.

Assume that there were an entity-linking specification $\mathcal{E}'$ in the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ of maximum-value solutions that is certain-link equivalent to $\mathcal{E}$; we shall derive a contradiction.

We denote by $\mathcal{E}' \restriction L_1$ the entity-linking specification that consists of the matching constraint for L_1, the FDs for L_1, and the inclusion dependencies for L_1.

Because of I_1, the inclusion dependencies for L_1 are $L_1[X| \subseteq D$ and $L_1[Y| \subseteq D$. From the inclusion dependencies, we see that $L_1(0,0)$ is the only candidate link for L_1 in I_1, I_2, I_3, and I_4. Throughout the rest of this proof, we will focus our attention only on this link $L_1(0,0)$. So we assume throughout the rest of this proof that $x = y = 0$.

If in $\mathcal{E}' \restriction L_1$, there is no matching constraint, then $L(0,0)$ would be certain link in I_2, a contradiction.

Assume that in $\mathcal{E}' \restriction L_1$, we have that the matching constraint for L_1 is

$$L_1(x, y) \rightarrow \forall \mathbf{u}(\psi(x, y, \mathbf{u}) \rightarrow \alpha_1 \vee \ldots \vee \alpha_k), \tag{5}$$

where $\psi(x, y, \mathbf{u})$ is a conjunction (possibly empty) of atomic formulas, where the universally quantified variables $\mathbf{u}$ must occur in ψ, and where α_i is of the form $\exists \mathbf{z_i}\, \phi_i(x, y, \mathbf{u}, \mathbf{z_i})$. Each ϕ_i is a conjunction of atomic formulas and equalities. We assume that the variables in $\mathbf{z_i}$ are disjoint from the variables in ψ and from $\{x, y\}$.

We begin by considering the case where ψ is empty. Thus, in this case the matching constraint is of the form

$$L_1(x, y) \rightarrow (\alpha_1 \vee \ldots \vee \alpha_k). \tag{6}$$

Since $L_1(0,0)$ is a certain link for I_1, it satisfies the matching constraint with respect to I_1. Since $I_1 \subset I_2$, it follows by monotonicity that $L_1(0,0)$ satisfies the matching constraint with respect to I_2 (this monotonicity property when the matching constraint is of the form (6) was noted in the proof of Theorem 3.5 in [9]). Since $L_1(0,0)$ is the only candidate link for L_1 in I_2, and it satisfies the matching constraint and the inclusion dependencies, it follows that $L_1(0,0)$ is a certain link for I_2, a contradiction.

So the matching constraint for L_1 is of the form (5) where ψ is nonempty. If P, R, or T were to appear in ψ, then (5) would be trivially satisfied for I_2. Since $L_1(0,0)$ would satisfy the matching constraint, it would follow, as before, that $L_1(0,0)$ would be a certain link for I_2, a contradiction. Therefore, only D and S can appear in ψ. When the variables in $\mathbf{u}$ (if any) are all assigned the value 0 (under our assumption throughout this proof that $x = y = 0$), then ψ holds in I_3. Since I_3 satisfies (5), and since ψ holds in I_3 when all of the variables in $\mathbf{u}$ (if any) are assigned the value 0, it follows that there is some α_i that is satisfied in I_3 when all of the variables in $\mathbf{u}$ (if any) are assigned the value 0. Now the only predicates that can appear in α_i are D, S, and R, since P and T are empty in I_3.

We showed that $\psi \rightarrow \alpha_i$ is true in I_3 when all of the variables in $\mathbf{u}$ (if any) are assigned the value 0. Since the premise ψ can be satisfied in I_3 only when all of the variables in $\mathbf{u}$ are assigned the value 0, it follows that $\forall \mathbf{u}(\psi \rightarrow \alpha_i)$ is satisfied in I_3. Now the only predicates that can appear in $\forall \mathbf{u}(\psi \rightarrow \alpha_i)$ are D, S, and R, since ψ can contain only D and S, and the only predicates that can appear in α_i are D, S, and R. Therefore, since I_3 and I_4 agree on D, S, and R, we have that $\forall \mathbf{u}(\psi \rightarrow \alpha_i)$ holds in I_4. So (5) holds in I_4 (for $x = y = 0$), and hence, as before, $L_1(0,0)$ is a certain link for I_4, a contradiction. □

In summary, this section focused on comparisons of the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ of maximum-value solutions with the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{W}_\mathbf{1})$ of maximum-cardinality repairs and with the entity-linking framework $(\mathcal{L}_\mathbf{c}, \mathcal{S}_\mathbf{1}, \mathcal{V}_\mathbf{1})$ for recursion-free collective entity linking. Theorems 2 and 3 establish that $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ and $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{W}_\mathbf{1})$ are incomparable in terms of expressive power, while Theorem 4 establishes that $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ is strictly subsumed by $(\mathcal{L}_\mathbf{c}, \mathcal{S}_\mathbf{1}, \mathcal{V}_\mathbf{1})$.

5. Adding preference constraints

Staworko et al. [28] introduced the idea of preferring some repairs over others. In this section, we introduce a family of entity-linking frameworks $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{P}_\mathbf{\Pi})$ that is parameterized by a set of Π *preference constraints*. This family of frameworks can be seen as an extension of the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$, where we use a more refined collection of weight functions that also take into account preferences among the link facts.

We first introduce the language of preference constraints from which Π is drawn. The main motivation for such preference constraints is that they allow a user to specify explicitly whether some link facts should be considered stronger than other link facts. Such preference constraints are given independently of, and in addition to, the set Σ of constraints in

$\mathcal{S}_0$, and will be used to further differentiate among conflicting links (i.e., pairs of link facts that violate one or both of the functional dependencies on a link relation).

A *preference constraint* has the following general form:

$$L(x, y) \wedge L(x', y') \wedge \alpha(x, y) \wedge \neg\alpha(x', y') \rightarrow L(x, y) \geq L(x', y') \tag{7}$$

In the above, L can be any of the link symbols in $\mathbf{L}$ while $\alpha(x, y)$ can be any predicate of the form $\exists \mathbf{z}\, \phi(x, y, \mathbf{z})$, where ϕ is a conjunction of source atomic formulas along with equalities.

Example 4. Consider a variation of the earlier Example 1 linking subsidiaries with companies, where the set Σ of constraints is as follows. The functional and inclusion dependencies are as before. However, the matching constraint is simplified, for the purposes of this example, so that it now requires only the similarity of the subsidiary name and company name:

$$L(sid, cid) \rightarrow \forall sn, loc, cn, hd\ (\texttt{Subsid}(sid, sn, loc) \wedge \texttt{Company}(cid, cn, hd) \rightarrow (sn \sim cn).$$

We now consider, additionally, a set Π consisting of a single preference constraint, which uses an `Exec`-based condition to differentiate among links:

$$\begin{aligned} &L(sid, cid) \wedge L(sid', cid') \\ &\quad \wedge \exists e, n, t, sn, loc\ (\texttt{Exec}(e, cid, n, t) \wedge \texttt{Subsid}(sid, sn, loc)\ \wedge \text{contains}(t, sn)) \\ &\quad \wedge \neg\exists e, n, t, sn, loc\ (\texttt{Exec}(e, cid', n, t) \wedge \texttt{Subsid}(sid', sn, loc)\ \wedge \text{contains}(t, sn)) \\ &\quad \rightarrow L(sid, cid) \geq L(sid', cid') \end{aligned}$$

Thus, whenever we have two links relating a subsidiary with a company, if one of the links satisfies the fact that the company has an executive whose title contains the subsidiary name, while the other link does not satisfy such fact, we prefer the first link over the second link.

Note that a user has the freedom, in general, to choose which conditions to push into the matching constraints of Σ and which ones into the preference constraints of Π. This is manifested, in this example, via the fact that the executive information is used in a preference constraint whereas before it was used as part of a matching constraint.

The notion of a consistent instance when there are preference constraints continues to be the same as that of a consistent instance with respect to an entity-linking specification in $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ where there are no preference constraints. Thus, the set Π of preference constraints plays no role in defining consistent instances under $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{P}_\mathbf{\Pi})$. However, Π plays an important role in defining the weight functions for the links, as we see next.

We are now ready to formally define $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{P}_\mathbf{\Pi})$. First, we recall from Section 3.2 the instance $\langle I, I^*\rangle$, which for a given source instance I, represents a superset for the repairs that we consider. Thus, I^* represents the domain for all the links that may appear in link relations.

Framework 6. The family of entity-linking frameworks $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{P}_\mathbf{\Pi})$ with preference constraints.

For every fixed finite set Π of preference constraints, we define an entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{P}_\mathbf{\Pi})$, by assigning to each $\Sigma \in \mathcal{S}_0$ a weight function $w_{\Sigma,\Pi}$ that depends on both Σ and Π. Given an $\mathbf{R}$-instance $\langle I, J\rangle$ and a fact $L^J(a, b)$, we define $w_{\Sigma,\Pi}(\langle I, J\rangle, L^J(a, b))$ to be $w_{\Sigma,\Pi}(\langle I, I^*\rangle, L^{I^*}(a, b))$, which in turn is defined as follows.

For each link symbol L, and source instance I, we first compute a preference relation $\geq_L$ on I^* on conflicting links of L, by evaluating each preference constraint of the form (7) that involves L. Concretely, whenever (x_0, y_0) and (x'_0, y'_0) are pairs in I^* such that $L(x_0, y_0)$ and $L(x'_0, y'_0)$ are conflicting (i.e., together violate one or both of the functional dependencies on L), and such that $\alpha(x_0, y_0)$ is true in I but $\alpha(x'_0, y'_0)$ is not true in I, we set $L(x_0, y_0) \geq_L L(x'_0, y'_0)$. In general, $\geq_L$ can have cycles. For example, we can have two distinct pairs $l = L(x_0, y_0)$ and $l' = L(x'_0, y'_0)$ such that $l \geq_L l'$ and $l' \geq_L l$. Such situation may arise when a user gives (at least) two preference constraints for L, the evaluation of which leads to opposite preferences for the particular links.

We then turn $\geq_L$ into an acyclic relation $>_L$ as follows. First, we take the transitive closure $\geq_L^*$ of $\geq_L$. Then, we set $l >_L l'$ whenever $l \geq_L^* l'$ but it is not the case that $l' \geq_L^* l$. Intuitively, $l >_L l'$ means that l is strictly preferred to l'. It can be verified that, for each L, the relation $>_L$ (or rather its inverse $<_L$) forms a strict partial order. We may also drop the subscript L and use the notation $>$ or $(\geq)$ whenever L is understood from the context. We may refer to $>$ as the *preference relation*.

The weight of a link fact l in I^* is then defined recursively as follows:

$$\begin{aligned} &w_{\Sigma,\Pi}(\langle I, I^*\rangle, l) = w_\Sigma(\langle I, I^*\rangle, l), \quad \text{if there is no } l' \text{ such that } l > l'; \\ &w_{\Sigma,\Pi}(\langle I, I^*\rangle, l) = w_\Sigma(\langle I, I^*\rangle, l) + \sum_{l>l'} w_{\Sigma,\Pi}(\langle I, I^*\rangle, l'), \quad \text{otherwise.} \end{aligned}$$

In the above, w_Σ is the weight function associated with Σ in the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ of maximum-value solutions. Thus, the weight of l is obtained by adding up $w_\Sigma(\langle I, I^*\rangle, l)$, which is calculated solely based on Σ as defined

for $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$, with the total aggregated weight of all the links that l dominates (via the preference relation $>$). In the special case when there are no preference constraints, the weight of a link l falls back to $w_{\Sigma}(\langle I, I^*\rangle, l)$. Thus, for each Π, the entity-linking framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{P}_{\mathbf{\Pi}})$ is an extension of the entity-linking framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$.

Note that, by definition, the weight of a link is relative to $\langle I, I^*\rangle$, on which we evaluated the preference constraints, but independent of any particular sub-instance $\langle I, J\rangle$. Thus, the hypothesis of Theorem 1 holds, by definition.

Example 5. Recall the specification in Example 4. First, it is immediate to see that this is an example of an entity-linking specification in the entity-linking framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{P}_{\mathbf{\Pi}})$, for the given set Π of preference constraints. Moreover, let us assume the same source instance I as in Example 2. The link $L(s_1, c_1)$ strictly dominates the link $L(s_1, c_2)$ (by the fact that c_1 satisfies the `Exec` condition for s_1 in the preference constraint, while c_2 does not). Since no other strict domination holds, we have that $w_{\Sigma,\Pi}(\langle I, I^*\rangle, L^{I^*}(s_1, c_1)) = 2$, while the weight of any other link is 1. As a consequence, among the four maximal cardinality repairs for $\langle I, I^*\rangle$ that we have seen earlier, we have that $\langle I, J_1\rangle$ and $\langle I, J_2\rangle$ have weight 3, while $\langle I, J_3\rangle$ and $\langle I, J_4\rangle$ have weight 2. Thus, $\langle I, J_1\rangle$ and $\langle I, J_2\rangle$ are the maximum weight repairs with respect to Σ and $w_{\Sigma,\Pi}$. As a result, we also obtain that $L(s_1, c_1)$ is the sole certain link, in this example.

As we noted above, the hypothesis of Theorem 1 holds for $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{P}_{\mathbf{\Pi}})$ and so we obtain, as a corollary, a polynomial-delay algorithm for the enumeration of maximum weight repairs and a polynomial-time algorithm for the computation of the certain links.

It is clear that every entity-linking framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$ (Framework 1) can be simulated by using an entity-linking framework involving preferences (Framework 6) by simply taking the set Π of preferences to be empty. The next theorem says that, in fact, we gain expressive power by allowing preference constraints. This is our main technical result.

Theorem 5. *There is a finite set Π of preference constraints such that the corresponding framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{P}_{\mathbf{\Pi}})$ is not subsumed by the entity-linking framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$ of maximum-value solutions.*

A key tool in the proof of Theorem 5 is a locality theorem that is interesting in its own right, and that we use multiple times in the proof of Theorem 5. We note that locality theorems have been used extensively in finite model theory to obtain inexpressibility results [25].

We begin with some preliminaries. For each entry a in a fact in an instance I, define $N_0^I(a)$ to be $\{a\}$. Inductively, define $N_{i+1}^I(a)$ to consist of $N_i^I(a)$ along with each c such that there is a' in $N_i^I(a)$ where a' and c are both entries in some fact in I. Thus $N_r^I(a)$ consists of those entries of I within distance r of a in the Gaifman graph [25] of I. Let $N_r^I(a, b)$ be $N_r^I(a) \cup N_r^I(b)$. We may refer to $N_r^I(a, b)$ as an *r-neighborhood*. We are interested in $N_r^I(a, b)$ only for source instances I. When I is understood, we may write simply N_r rather than N_r^I.

In the statement of the Locality Theorem, by $I \restriction N_r^I(a_i, b_i)$ we mean the usual notion of the restriction of I to the domain $N_r^I(a_i, b_i)$.

Theorem 6 *(Locality Theorem). Let $\mathcal{E}$ be an entity-linking specification in $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$, with link symbol L. Then there is r, depending only on $\mathcal{E}$, such that for every source instance I, every link instance J, and every a_1, b_1, a_2, b_2 in I, if $I \restriction N_r^I(a_1, b_1)$ and $I \restriction N_r^I(a_2, b_2)$ are isomorphic under an isomorphism f with $f(a_1) = a_2$ and $f(b_1) = b_2$, then the weights of the links $L(a_1, b_1)$ and $L(a_2, b_2)$ in $\mathcal{E}$ are the same, that is, $w(\langle I, J\rangle, L^J(a_1, b_1)) = w(\langle I, J\rangle, L^J(a_2, b_2))$.*

Proof. If there is no matching constraint for L, then we can take $r = 0$, since the weight of every link is then 1. So assume that the link L has the matching constraint (1). By the Gaifman locality theorem for first-order logic [17], we have that there is r' such that if $I \restriction N_{r'}^I(a_1, b_1)$ and $I \restriction N_{r'}^I(a_2, b_2)$ are isomorphic under an isomorphism f with $f(a_1) = a_2$ and $f(b_1) = b_2$, then we have that $\exists \mathbf{u}\psi(x, y, \mathbf{u})$ holds when $x = a_1$ and $y = b_1$ if and only if $\exists \mathbf{u}\psi(x, y, \mathbf{u})$ holds when $x = a_2$ and $y = b_2$. Furthermore, by the Gaifman locality theorem for first-order logic with counting [24], it follows that for each positive integer c, there is r with $r \geq r'$ such that if $I \restriction N_r^I(a_1, b_1)$ and $I \restriction N_r^I(a_2, b_2)$ are isomorphic under an isomorphism f with $f(a_1) = a_2$ and $f(b_1) = b_2$, then we have that

$$\exists \mathbf{u}(\psi(x, y, \mathbf{u}) \wedge (\bigvee_{c_1,\ldots,c_k, \text{s.t.} c = c_1 + \cdots + c_k} \#\mathbf{z_1}.\phi_1(x, y, \mathbf{u}, \mathbf{z_1}) = c_1 \wedge \cdots \wedge \#\mathbf{z_k}.\phi_k(x, y, \mathbf{u}, \mathbf{z_k}) = c_k)$$

holds when $x = a_1$ and $y = b_1$ if and only if it holds when $x = a_2$ and $y = b_2$. Here α_i in (1) is $\exists \mathbf{z_i}\, \phi_i(x, y, \mathbf{u}, \mathbf{z_i})$, and $\#\mathbf{z_i}.\phi_i(x, y, \mathbf{u}, \mathbf{z_i})$ is a count of the number of tuples $\mathbf{z_i}$ that satisfy $\phi_i(x, y, \mathbf{u}, \mathbf{z_i})$. In fact, it turns out that the same value of r can be used for each choice of c: this follows from results by Libkin [24], which say that the choice of r depends only on the quantifier rank of the formula, and so is independent of c.

These observations imply the theorem, based on the definition of the weight of a link in $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$. □

We now give a sketch of the proof of Theorem 5. We give the proof in full in the appendix.

Please cite this article in press as: D. Burdick et al., Expressive power of entity-linking frameworks, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.09.001

Sketch of the proof of Theorem 5. Our entity-linking specification $\mathcal{E}$ in the framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{P}_{\mathbf{\Pi}})$ has one link symbol L, the matching constraint $L(x, y) \rightarrow R(x, y)$, both FDs on L, and the inclusion dependencies $L[X| \subseteq R[X]$ and $L[Y| \subseteq R[Y]$. We define a family of source instance K_r and a set of preference constraints such that we get two long chains $L(0, 1) > L(2, 3) > L(4, 5) > \cdots > L(m, m+1)$ and $L(0, 1') > L(2', 3') > L(4', 5') > \cdots > L(n', (n+1)')$ of strict preferences, where $m > n$ (so the first chain is longer than the second). It is shown that $L(0, 1)$ has so much weight that it is a certain link for $\mathcal{E}$. However, given an entity-linking specification $\mathcal{E}'$ in the entity-linking framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$ of maximum-value solutions, when we select r based on $\mathcal{E}'$, the source instance $K = K_r$ is designed so that the neighborhoods $K \restriction N_r^K(0, 1)$ and $K \restriction N_r^K(0, 1')$ are isomorphic, and so by the Locality Theorem, $L(0, 1)$ and $L(0, 1')$ have the same weight in $\mathcal{E}'$.

Assume, by way of contradiction, that there is an entity-linking specification $\mathcal{E}'$ in the entity-linking framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$ that is certain-link equivalent to $\mathcal{E}$. By considering an instance with only one fact $R(0, 1)$, we show that $\mathcal{E}'$ has the same inclusion dependencies as $\mathcal{E}$. We show that $\mathcal{E}'$ has both FDs on L with the following argument. Assume first that $\mathcal{E}'$ does not have the FD $L: \ Y \rightarrow X$. Since $L(0, 1')$ has the same weight in $\mathcal{E}'$ as $L(0, 1)$, in particular $L(0, 1')$ satisfies the matching constraint for $\mathcal{E}'$. Now $L(0, 1')$ is not a certain link in $\mathcal{E}'$, since it is not a certain link in $\mathcal{E}$. So let $\langle K, N\rangle$ be a maximum weight repair of $\langle K, K^*\rangle$ that does not contain $L(0, 1')$. Then of course N contains the certain link $L(0, 1)$. Form N' by replacing $L(0, 1)$ in N by $L(0, 1')$. Now N' satisfies the only possible FD $L: \ X \rightarrow Y$, and it satisfies the inclusion dependencies and matching constraint. Furthermore, N' has the same weight as N, since $L(0, 1)$ and $L(0, 1')$ have the same weight, and so $\langle K, N'\rangle$ is a maximum weight repair. But this is a contradiction, since $\langle K, N'\rangle$ is a maximum weight repair that does not contain the certain link $L(0, 1)$. Now define the instance $U(K)$, where (a, b) is a tuple of a relation of K if and only if $(\underline{b}, \underline{a})$ is a tuple of the corresponding relation of $U(K)$, and where $\underline{a}$ and $\underline{b}$ are new values. The proof that the FD $L: \ X \rightarrow Y$ holds for $\mathcal{E}'$ is the same, except rather than replacing the certain link $L(0, 1)$ in a maximum weight repair of $\langle K, K^*\rangle$ by $L(0, 1')$, we instead replace the certain link $L(\underline{1}, \underline{0})$ in a maximum weight repair of $\langle U(K), (U(K))^*\rangle$ by $L(\underline{1'}, \underline{0})$.

We explicitly find the set M of certain links for $I = K \cup U(K)$ in $\mathcal{E}$ and prove, using the FDs and inclusion dependencies for $\mathcal{E}'$, that $\langle I, M\rangle$ is the unique maximum weight repair for $\langle I, I^*\rangle$ in $\mathcal{E}'$. Let M' consist precisely of all of the links of $\mathcal{E}$ that are not links in M. We prove, again using the Locality Theorem, that there is a one-to-one correspondence between the links ℓ of M and the links ℓ' of M', where ℓ and ℓ' have the same weight in $\mathcal{E}'$. In particular, each link of M' satisfies the entity-linking specification of $\mathcal{E}'$. Further, since M' also satisfies both FDs and the inclusion dependencies, it follows that $\langle I, M'\rangle$ is a maximum weight repair. But this is a contradiction, since $\langle I, M\rangle$ is the unique maximum weight repair. □

6. Concluding remarks

In this article, we introduced and explored a unifying approach to entity linking. This approach, which is based on the notion of an entity-linking framework and the notion of the certain links in such a framework, provides a single formalism for modeling different entity-linking scenarios and for comparing them using the certain links as a measure of their expressive power. To this effect, we defined a notion of *certain-link equivalence* that allows us to compare entity-linking specifications, in a way other than logical equivalence (which may be too strict for entity linking purposes). We then made use of certain-link equivalence to define what it means for an entire entity-linking framework to subsume another one. We established a number of technical results that delineate the comparative expressive power of several concrete entity-linking frameworks.

Our concrete focus in this article was on the comparison of the entity-linking framework of maximum-value solutions with entity-linking frameworks (1) that involve maximum cardinality repairs, (2) that allow recursion-free collective entity linking, and (3) that incorporate preferences among links. It might be interesting to compare the expressive power of other frameworks defined in this article.

There are several other directions of research that arise from the work reported here. To begin with, Theorem 1 gives a sufficient condition for the tractability of computing the certain links for a family of entity-linking frameworks based on $\mathcal{L}_{\mathbf{0}}$. The certain links are the consistent answers of the atomic queries involving a single link relation. A next step would be to investigate how the complexity changes for the case of consistent answers of more general queries involving link relations and source relations. Another next step is to understand the expressive power of recursive collective entity linking. Specifically, we conjecture that the framework $(\mathcal{L}_{\mathbf{c}}, \mathcal{S}_{\mathbf{2}}, \mathcal{V}_{\mathbf{2}})$ of recursive collective entity linking cannot be subsumed by the framework $(\mathcal{L}_{\mathbf{c}}, \mathcal{S}_{\mathbf{1}}, \mathcal{V}_{\mathbf{1}})$ of non-recursive collective entity linking. Another next step has to do with Markov Logic Networks (MLNs), which were first studied in [27]. As stated earlier, it follows from results in [9] that linear MLNs are subsumed by an entity-linking framework of maximum-value solutions with weighted disjuncts, where the constraints are in the existential fragment $\exists\mathcal{L}_{\mathbf{0}}$ of the language $\mathcal{L}_{\mathbf{0}}$. It is an open problem if more general MLNs (i.e., not necessarily linear) can be subsumed by an entity-linking framework of maximum-value solutions with weighted disjuncts for some suitable choice of weights and constraints from $\mathcal{L}_{\mathbf{0}}$ or from the more general language $\mathcal{L}_{\mathbf{c}}$ of collective entity linking.

In a different direction, we note that our unifying approach to entity linking is flexible enough to allow assigning *probabilities* to links in a natural way. Specifically, we can define the probability $Pr(L(a, b))$ of a link $L(a, b)$ to be the number of maximum weight repairs containing $L(a, b)$ divided by the total number of maximum weight repairs. Thus, a link $L(a, b)$ is certain if and only if $Pr(L(a, b)) = 1$. The introduction of probabilities in entity-linking frameworks raises several algorithmic questions, including the question of enumerating the links whose probability is above a fixed threshold, say, enumerating all links $L(a, b)$ such that $Pr(L(a, b)) \geq 0.75$. Furthermore, it may be possible to establish tight connections between our approach and other approaches in entity linking and entity resolution, such as Probabilistic Soft Logic (PSL) [3,4,8], that derive

Please cite this article in press as: D. Burdick et al., Expressive power of entity-linking frameworks, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.09.001

links with *scores* based on weighted first-order formulas. By utilizing such connections, one may also be able to transfer the formalism of preference constraints, which fits naturally in our declarative approach, into PSL (or into MLN as well). In general, we may obtain more powerful entity linking approaches that combine declarative, logic-based specification with probabilistic reasoning and with explicit user preference constraints.

Acknowledgments

Part of this work was done while Phokion G. Kolaitis was visiting the Simons Institute for the Theory of Computing.

Appendix A. Proof of Theorem 3

Theorem 3. *The entity-linking framework* $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{W}_{\mathbf{1}})$ *of maximum cardinality repairs is not subsumed by the entity-linking framework* $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$ *of maximum-value solutions.*

Proof. We now give an entity-linking specification $\mathcal{E}$ in $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{W}_{\mathbf{1}})$ for which there is no $\mathcal{E}'$ in the framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$ of maximum-value solutions that is certain-link equivalent to it.

Let $\mathcal{E}$ be the entity-linking specification given by the following:

- $L(x, y) \rightarrow R(x, y) \vee S(x, y)$
- FD $L : X \rightarrow Y$
- $L[X] \subseteq D_1$
- $L[Y] \subseteq D_2$

Note that the relations D_1 and D_2, which appear in the inclusion dependencies, are unary.

For each of the instances I_j we now specify, the role of J in Definition 10 is played by $(I_j)^*$, which is defined in Definition 9.

For $I_1 = \{R(0, 1), S(2, 3), D_1(0), D_1(2), D_2(1), D_2(3)\}$, we have $L(0, 1)$ and $L(2, 3)$ as certain links.

For $I_2^p = \{S(0, 1), R(0, 2), S(0, 2), D_1(0), D_2(1), D_2(2), \ldots, D_2(p)\}$, where $p \geq 2$, we have no certain links, as we now show. Even though $L(0, 1)$ and $L(0, 2)$ satisfy the matching constraint and the inclusion dependencies, they each have weight 1 in the entity-linking framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{W}_{\mathbf{1}})$ of maximum cardinality repairs. Because of the FD, it then follows that neither is a certain link. For notational simplicity, denote I_2^p simply by I_2 when $p = 2$.

For $I_3^p = \{R(0, 2), D_1(0), D_2(1), D_2(2), \ldots, D_2(p)\}$, where $p \geq 2$, we have $L(0, 2)$ as a certain link. For notational simplicity, denote I_3^p simply by I_3 when $p = 2$.

For $I_4 = \{D_1(0), D_2(1)\}$, we have no certain link, as we now show. The only possible link based on the inclusion dependencies is $L(0, 1)$ but $L(0, 1)$ does not satisfy the matching constraint.

Assume that there were an entity-linking specification $\mathcal{E}'$ in the framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$ of maximum-value solutions that is certain-link equivalent to $\mathcal{E}$; we shall derive a contradiction.

Because of I_1, we see that $\mathcal{E}'$ has the inclusion dependencies $L[X] \subseteq D_1$ and $L[Y] \subseteq D_2$, since no projection of R or S contains all of the values of $L[X]$ (respectively, of $L[Y]$), and since D_2 does not contain all of the values of $L[X]$, and D_1 does not contain all of the values of $L[Y]$.

If $\mathcal{E}'$ has no matching constraint, then $L(0, 1)$ would be a certain link in I_4, because (1) it fulfills the inclusion dependencies, and (2) no FD is violated since $L(0, 1)$ is the only possible link because of the inclusion dependencies. But this is a contradiction, since $L(0, 1)$ is not a certain link in I_4.

Let the matching constraint for $\mathcal{E}'$ be of the form (1), with the same restrictions as given there. We begin by considering the case where ψ is empty. Thus, in this case the matching constraint is of the form (2).

We now show that as far as the links $L(x, y)$ that we consider in this portion of this proof (where we are assuming that ψ is empty), we can assume without loss of generality that no equality appears in any of the disjuncts. Let α_i be a disjunct. Form equivalence classes of the variables in α_i by putting two variables v and w in the same equivalence class precisely if there is a sequence of equalities in α_i that force v and w to be equal. If the distinguished variables x and y are in the same equivalence class, then that disjunct cannot be satisfied for the links $L(x, y)$ and instances considered in this portion of this proof, because we have disjoint domains for x and y; therefore, we can delete that disjunct. Otherwise, let x be the representative for the equivalence class containing x, let y be the representative for the equivalence class containing y, and for the other equivalence classes arbitrarily select a representative. Then replace each variable by the representative of its equivalence class, and if the existentially quantified variable z is thereby eliminated, delete $\exists z$ from the disjunct. These changes have no effect on the weight of a disjunct when we are calculating the weight of a link, since (a) when we eliminate a disjunct because it has x and y in the same equivalence class, that disjunct could not be satisfied anyway for the links and instances considered in this proof, and so that disjunct would not contribute to the weight of a link, and (b) the weight is determined by the number of satisfying truth assignments to the existentially quantified variables, and for each choice for the element assigned to the representatives, there is a unique choice for the element assigned to the other variables, so these changes do not affect number of satisfying truth assignments.

Since $L(0,2)$ is a certain link in I_3 and $L(0,1)$ is not, there must be some disjunct α_{i_0} that distinguishes between $y=1$ and $y=2$, and that is satisfied in I_3 when $x=0$ and $y=2$. Since no disjunct contains an equality, and since neither $D_1(y)$ nor $D_2(y)$ distinguishes between $y=1$ and $y=2$, the only feasible possibility is $R(v,y)$ where v is either x or an existentially quantified variable. So there is some disjunct α_{i_0} that is satisfied in I_3 when $x=0$ and $y=2$ and where α_{i_0} contains $R(v,y)$ for some variable v.

In I_2, we see from the inclusion dependencies that the only possible links are $L(0,1)$ and $L(0,2)$. We now show that for each disjunct α_i, we have that $L(0,2)$ gets at least the same weight in I_2 from α_i as $L(0,1)$. This is certainly true if α_i contributes 0 to $L(0,1)$. So assume that α_i contributes a positive weight to $L(0,1)$. Let α_i be $\exists \mathbf{z_i} \phi_i$, and let t be an arbitrary assignment to the variables of $\mathbf{z_i}$ that makes ϕ_i hold in I_2 when $x=0$ and $y=1$. We now show that t also makes ϕ_i hold in I_2 when $x=0$ and $y=2$. Since t makes ϕ_i hold in I_2 when $x=0$ and $y=1$, the only possible appearances of y in ϕ_i are as either $D_2(y)$ or as $S(v,y)$ for some variable v (where v is either x or some existentially quantified variable). In both cases, the same atomic formula holds in I_2 under t when $x=0$ and $y=2$. Since the weight for α_i is the sum over all assignments t that make ϕ_i hold in I_2, it follows that $L(0,2)$ gets at least the same weight in I_2 from α_i as $L(0,1)$.

Now let α_{i_0} be as defined above. In particular, α_{i_0} is satisfied in I_3 when $x=0$ and $y=2$. Then α_{i_0} is also satisfied in I_2 when $x=0$ and $y=2$, since I_3 is a sub-instance of I_2. However, α_{i_0} is not satisfied when $x=0$ and $y=1$ in I_2, since α_{i_0} contains $R(v,y)$ for some variable v, and $R(v,y)$ does not hold in I_2 for any choice of v when $y=1$.

We have shown that in I_2, every disjunct that gives a positive weight to $L(0,1)$ gives at least that same positive weight to $L(0,2)$, and that in addition there is a disjunct α_{i_0} that gives a positive weight to $L(0,2)$ but not to $L(0,1)$. Therefore, $L(0,2)$ has a strictly higher weight in I_2 than $L(0,1)$. Because of the inclusion dependencies, the only two possible inks for I_2 are $L(0,1)$ and $L(0,2)$. Therefore, $L(0,2)$ should be a certain link in I_2. But it is not. This contradiction completes the case when $\mathcal{E}'$ has the matching constraint (1) with ψ empty.

Now consider the other case, where $\mathcal{E}'$ has the matching constraint (1), and where ψ is a nonempty conjunction of atomic formulas. We now show that ψ can contain only D_1 and/or D_2, but not R or S. For if ψ were to contain R or S, then in I_4 the matching constraint for L would be trivially satisfied when $x=0$ and $y=1$. Since the inclusion dependencies are satisfied in I_4 when $x=0$ and $y=1$, and since the inclusion dependencies also tell us that $L(0,1)$ is the only possible link for I_4, this would imply that $L(0,1)$ is a certain link for I_4, which is a contradiction. So indeed, ψ can contain only D_1 and/or D_2.

We now show that we can assume without loss of generality that no disjunct α_i contains an equality involving an existentially quantified variable in $\mathbf{z_i}$. If an equality $z=z'$ appears in α_i where z and z' are in $\mathbf{z_i}$, then pick one of z and z', say z', and replace every occurrence of z' by z and remove the existential quantifier $\exists z'$. After this, if an equality $x=z$ (or $z=x$) occurs in α_i, then replace every occurrence of z by x and remove the existential quantifier $\exists z$. After this, if an equality $y=z$ (or $z=y$) occurs in α_i, then replace every occurrence of z by y and remove the existential quantifier $\exists z$. After this, if an equality $u=z$ (or $z=u$) occurs in α_i, where u is a universally quantified variable in $\mathbf{u}$, then replace every occurrence of z by u and remove the existential quantifier $\exists z$. As before, a uniqueness argument shows that these changes have no effect on the weights of links.

Furthermore, we can assume without loss of generality that no α_i contains an equality $v=v$ (that is, an equality where the left-hand side and the right-hand side are the same), since this can have no effect on distinguishing the weights of links.

We have shown that ψ can contain only D_1 and/or D_2. If ψ contains only D_1, then, since for the links and instances we consider in this portion of this proof, the D_1-relation contains precisely one entry (namely, 0), it follows that we are effectively back to the case we already considered, where ψ is empty. So assume that ψ contains at least one occurrence of D_2. Let us refer to each universally quantified variable u such that $D_1(u)$ appears in ψ as a U_1-*variable*, and each universally quantified variable u such that $D_2(u)$ appears in ψ as a U_2-*variable*. If some variable were both a U_1-variable and a U_2-variable, then the matching constraint would be trivially satisfied in I_4 when $x=0$ and $y=1$, and so as before, $L(0,1)$ would be a certain link for I_4, a contradiction. So every universally quantified variable is a U_1-variable or a U_2-variable but not both. For the links and instances we consider in this portion of this proof, we can assume without loss of generality that no α_i contains $D_1(y)$ or $D_2(x)$, or contain $D_1(u)$ for a U_2-variable u, or contain $D_1(u)$ for a U_2-variable u, as we now explain. Assume first that α_i were to contain $D_1(y)$. Because of the inclusion dependencies, we know that $D_2(y)$ holds. For each of the links and instances considered in this proof, we have D_1 and D_2 disjoint. Therefore, if $D_1(y)$ were to appear in α_i, we know that α_i would fail, and so contribute nothing to the weights of a link. So assuming that α_i does not contain $D_1(y)$ has no effect on the weights of links in this proof. Therefore, in this proof, we can assume without loss of generality that no α_i contains $D_1(y)$. Similarly, we can assume without loss of generality that no α_i contains any of $D_2(x)$, $D_1(u)$ for a U_2-variable u, or $D_1(u)$ for a U_2-variable u.

Let m be the number of U_2-variables in the matching constraint. Let us now consider I_3^{m+2} (that is, I_3^p where $p=m+2$). Based on the inclusion dependencies, the only possible links are $L(0,j)$, for $1 \leq j \leq m+2$. Since $L(0,2)$ is the only certain link, it must be that $L(0,2)$ has a strictly higher weight than each $L(0,j)$ for $j \neq 2$. In particular, $L(0,2)$ has a strictly higher weight than $L(0,1)$. The weight for $L(0,1)$ is, by definition, the min over all choices of assignments t to the universally quantified variables of the weight for $\alpha_1 \vee \ldots \vee \alpha_k$ in I_3^{m+2} when $x=0$, $y=1$, and when the universally quantified variables are assigned according to t. Let t_0 be the assignment to the universally quantified variables where each U_1-variable is assigned 0 (the only possible choice in I_3), and the m U_2-variables are each assigned a different member of $\{3,\ldots,m+2\}$. The min occurs when the assignment to the universally quantified variables is t_0, since (a) for this choice of assignment

Please cite this article in press as: D. Burdick et al., Expressive power of entity-linking frameworks, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.09.001

to the universally quantified variables, no equality $u_1 = u_2$ holds for distinct U_2-variables u_1 and u_2, (b) we do not have $y = u$ holding for any universally quantified variable u, and (c) for both $j = 1$ and $j = 2$, every atomic formula involving a U_j-variable u that holds under the assignment t_0 also holds for every other assignment of u to a member of D_j. Hence, the weight for $L(0, 1)$ is the weight for $\alpha_1 \vee \ldots \vee \alpha_k$ when $x = 0$, $y = 1$, and the universally quantified variables have the assignment t_0. Similarly, the weight for $L(0, 2)$ is the weight for $\alpha_1 \vee \ldots \vee \alpha_k$ when $x = 0$, $y = 2$, and the universally quantified variables have the assignment t_0.

We now show that if α_i holds in I_3^{m+2} both (a) for $y = 1$ with the assignment t_0 to the universally quantified variables, and (b) for $y = 2$ with the assignment t_0 to the universally quantified variables, then α_i has the same weight in both cases. This is because an existentially quantified variable z can appear in α_i in only 4 contexts: either (i) $D_1(z)$ (where there is only one choice for z, namely 0), (ii) $D_2(z)$ (where there are $m + 2$ choices for z, and if one choice for z succeeds, then so do the other choice for z), (iii) $R(z, v)$ for some v (where there is only one choice for z, namely 0), or (iv) $R(v, z)$ for some v (where there is only one choice for z, namely 2).

From what we just showed, and from the fact that $L(0, 2)$ has a strictly higher weight in I_3^{m+2} than $L(0, 1)$, there must be some disjunct α_{i_0} that has a positive weight under t_0 when $y = 2$ but has weight 0 under t_0 when $y = 1$. Now α_{i_0} contains no equalities involving y, since (1) $y = u$ is false under t_0 for each universally-quantified variable u, as is $y = x$, and (2) we showed that we can assume that there are no equalities involving existentially-quantified variables and no equality $y = y$. We know that α_{i_0} must contain y in some context other than $D_2(y)$, or else α_{i_0} would take on the same weight for $y = 1$ and $y = 2$. The only feasible choice is $R(v, y)$ where v is either x or a U_1-variable or an existentially quantified variable. So α_{i_0} contains $R(v, y)$ where v is either x or a U_1-variable or an existentially quantified variable.

Now let us consider I_2^{m+2}. Similarly to before, the weights for $L(0, j)$, for $1 \leq j \leq m+2$, are obtained when the universally quantified variables have the assignment t_0. Assume that some disjunction α_i holds for $x = 0$, $y = j$ (for some j), and the assignment t_0. Now α_i is $\exists \mathbf{z}_i \phi_i$. Let t' be an assignment to the variables $\mathbf{z}_i$ that makes ϕ_i hold in I_2^{m+2} when $x = 0$, $y = j$, under the assignment t_0 to the universally quantified variables. Let t'_0 be the result of extending t_0 to include the assignment t'. We now show that ϕ_i holds in I_2^{m+2} also when $x = 0$ and $y = 2$ under the assignment t'_0. To see this, let us consider the syntactically possible contexts in which y can appear in ϕ_i.

- It cannot appear in an equality, as we noted.
- It can appear in the form $D_1(y)$.
- It can appear in the form $D_2(y)$.
- It can appear in the form $R(v, y)$ where v is a variable.
- It can appear in the form $R(y, v)$ where v is a variable.
- It can appear in the form $S(v, y)$ where v is a variable.
- It can appear in the form $S(y, v)$ where v is a variable.

In all of these cases, if ϕ_i holds in I_2^{m+2} when $x = 0$ and $y = j$, under the assignment t'_0, then ϕ_i holds in I_2^{m+2} when $x = 0$ and $y = 2$, under the assignment t'_0.

Therefore, α_i (in particular, when $i \neq i_0$) has at least the same positive weight when $x = 0$ and $y = 2$ under the assignment t_0 as when $x = 0$ and $y = j$ under the assignment t_0. We now show that there is an additional positive weight, obtained from α_{i_0}, when $x = 0$ and $y = 2$, but no additional positive weight when $x = 0$ and $y = j$ for $j \neq 2$. Since (a) α_{i_0} holds in I_3^{m+2} when $x = 0$, $y = 2$, and when the universally quantified variables are assigned values according to t_0, and (b) I_3^{m+2} is a sub-instance of I_2^{m+2}, it follows that α_{i_0} holds in I_2^{m+2} when $x = 0$, $y = 2$, and the universally quantified variables are assigned values according to t_0. So α_{i_0} gives positive weight when $x = 0$, $y = 2$, and the universally quantified variables are assigned values according to t_0. However, α_{i_0} gives no positive weight when $x = 0$ and $y = j$ for $j \neq 2$, since α_{i_0} contains $R(v, y)$ for some variable v, and $R(a, j)$ does not hold in I_2^{m+2} for any a.

Let s_j be the weight for $\alpha_1 \vee \ldots \vee \alpha_k$ in I_2 when $x = 0$ and $y = j$ and the universally quantified variables are assigned values according to t_0. We just showed that $s_2 > s_j$ for $j \neq 2$. But from what we said before, we know that s_j is the weight for $L(0, j)$ in I_2^{m+2}. Therefore $L(0, 2)$ has a strictly higher weight in I_2^{m+2} than $L(0, j)$ for $j \neq 2$. Hence, $L(0, 2)$ is a certain link for I_2^{m+2}, which is a contradiction. □

Appendix B. Proof of Theorem 5

Theorem 5. *There is a finite set Π of preference constraints such that the corresponding framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{P}_{\mathbf{\Pi}})$ is not subsumed by the entity-linking framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$ of maximum-value solutions.*

Proof. Because of the length and complexity of the proof, we break it into subsections. □

B.1. The instance $I_{n+4,n}$

Let Σ be the following constraints:

- $L(x, y) \rightarrow R(x, y)$
- FDs $L: X \rightarrow Y$ and $L: Y \rightarrow X$
- $L[X| \subseteq R[X]$
- $L[Y| \subseteq R[Y]$

Further, we take the following set Π of three preference constraints:

$$L(x, y) \wedge L(x', y') \wedge W_i(x, y) \wedge \neg W_i(x', y') \rightarrow L(x, y) \geq L(x', y'),$$

for $1 \leq i \leq 3$.

Let $\mathcal{E}$ be the entity-linking specification given by the constraints Σ and the preferences Π.

Define the instance I_m where m is a positive integer that is a multiple of 4, to consist of the following facts:

- $R(2i, 2i+1)$ for $0 \leq i \leq m$
- $R(2i, 2i-1)$ for $1 \leq i \leq m$
- $W_1(4i, 4i+1)$ for $0 \leq i \leq \frac{m}{4}$
- $W_1(4i+2, 4i+1)$ for $0 \leq i \leq \frac{m}{4} - 1$
- $W_1(4i, 4i-1)$ for $\frac{m}{4} + 1 \leq i \leq \frac{m}{2}$
- $W_1(4i, 4i+1)$ for $\frac{m}{4} + 1 \leq i \leq \frac{m}{2}$
- $W_2(4i, 4i+1)$ for $0 \leq i \leq \frac{m}{4}$
- $W_2(4i+2, 4i+3)$ for $0 \leq i \leq \frac{m}{4} - 1$
- $W_2(4i, 4i-1)$ for $\frac{m}{4} + 1 \leq i \leq \frac{m}{2}$
- $W_2(4i+2, 4i+1)$ for $\frac{m}{4} \leq i \leq \frac{m}{2} - 1$
- $W_3(4i+2, 4i+3)$ for $0 \leq i \leq \frac{m}{4} - 1$
- $W_3(4i, 4i-1)$ for $1 \leq i \leq \frac{m}{4}$
- $W_3(4i+2, 4i+1)$ for $\frac{m}{4} \leq i \leq \frac{m}{2} - 1$
- $W_3(4i+2, 4i+3)$ for $\frac{m}{4} \leq i \leq \frac{m}{2} - 1$

Define the instance I'_n just as we defined I_m, except that we replace m by n (another positive integer), and we make use of primed positive integers i' instead of the corresponding positive integer i. For example, the condition "$W_1(4i, 4i+1)$ for $0 \leq i \leq \frac{m}{4}$" would be replaced by "$W_1(0, 1')$ and $W_1((4i)', (4i+1)')$ for $0 < i \leq \frac{n}{4}$". So the only entry that I_m and I'_n have in common is 0.

Now define the instance $I_{m,n}$ to be the union of the facts in I_m and I'_n. We assume throughout this proof that $m = n + 4$, so that both m and n are positive integers divisible by 4. The instance K used in the rough sketch of the proof of Theorem 5 in the body of the paper is $I_{n+4,n}$.

B.2. Type of links

Let us now consider what the preferences are, based on the preference constraints. The only situation where links $L(a_1, b_1)$ and $L(a_2, b_2)$ conflict where $R(a_1, b_1)$ is a fact in I_m and $R(a_2, b_2)$ is a fact in I'_n is the conflict between $L(0, 1)$ and $L(0, 1')$. But the preference constraints give us no preference in this conflict.

Let us call a link $L(a, b)$, where $R(a, b)$ is a fact of $I_{m,n}$, a $(+, +, -)$ link if $W_1(a, b)$ holds (this is represented by the first $+$), $W_2(a, b)$ holds (this is represented by the second $+$), and $W_3(a, b)$ fails to hold (this is represented by the $-$). Assume throughout the rest of this paragraph for illustrative purposes that $m \geq 12$. Thus, $L(0, 1)$, $L(4, 5)$, and $L(8, 9)$ are $(+, +, -)$ links. Call a link $L(a, b)$ a $(+, -, -)$ link (or a link of type $(+, +, -)$) if $W_1(a, b)$ holds (this is represented by the $+$), $W_2(a, b)$ fails to hold (this is represented by the first $-$), and $W_3(a, b)$ fails to hold (this is represented by the second $-$). Thus, $L(2, 1)$, $L6, 5)$, and $L(10, 9)$ are $(+, -, -)$ links. Similarly, there are two other types of links: the $(-, +, +)$ links, which include $L(2, 3)$, $L(6, 7)$, and $L(10, 11)$, and the $(-, -, +)$ links, which include $L(4, 3)$, $L(8, 7)$, and $L(12, 11)$.

The preference constraints tell us that for two distinct links $L(a_1, b_1)$ and $L(a_2, b_2)$, we have $L(a_1, b_1) \geq L(a_2, b_2)$ if (1) $L(a_1, b_1)$ and $L(a_2, b_2)$ are conflicting (that is, either $a_1 = a_2$ or $b_1 = b_2$), (2) $L(a_1, b_1)$ is a (c_1, c_2, c_3) link (where each c_j is either $+$ or $-$), (3) $L(a_2, b_2)$ is a (d_1, d_2, d_3) link, and (4) for some j, we have that $c_j = +$ and $d_j = -$.

In the sequence of possible links, namely

$$L(0, 1), L(2, 1), L(2, 3), L(4, 3), \ldots, L(2m, 2m-1), L(2m, 2m+1), \tag{B.1}$$

the pattern $(+, +, -)$, $(+, -, -)$, $(-, +, +)$, $(-, -, +)$ of types repeats over and over, with one exception: after $L(m, m+1)$ (which is of type $(+, +, -)$) we skip the type $(+, -, -)$, so that the link following $L(m, m+1)$, namely $L(m+2, m+1)$, is of type $(-, +, +)$ instead of $(+, -, -)$. The next link $L(m+2, m+3)$ is of type $(-, -, +,)$, just like other links that follow a link of type $-, +, +)$. Then the pattern $(+, +, -)$, $(+, -, -)$, $(-, +, +)$, $(-, -, +)$ of types again repeats over and over (ending with a link of type $(+, -, -)$). The same comment about the pattern of types, except with m replaced by n, applies to the sequence

Please cite this article in press as: D. Burdick et al., Expressive power of entity-linking frameworks, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.09.001

$$L(0,1'), L(2',1'), L(2',3'), L(4',3'), \ldots, L((2n)',(2n-1)'), L((2n)',(2n+1)'). \tag{B.2}$$

B.3. The sets S_i and S'_i

Define the set S_1 to consist of the single link $L(0,1)$; the set S_i to consist of the two links $L(2i-2,2i-3)$ and $L(2i-2,2i-1)$ for $1 < i \leq \frac{m}{2}$; the set $S_{\frac{m}{2}+1}$ to consist of the three links $L(m,m-1)$, $L(m,m+1)$ and $L(m+2,m+1)$; the set S_i to consist of the two links $L(2i-2,2i-1)$ and $L(2i,2i-1)$ for $\frac{m}{2}+2 < i \leq m$; and S_{m+1} to consist of the single link $L(2m,2m+1)$. For the sets S_i with more than one member, we refer to the member of S_i that appears first in (B.1) as the *first member* or *top member* of S_i, the member of S_i that appears second in (B.1) as the *second member* of S_i, and (in the case $i = \frac{m}{2}+1$, where S_i has three members) the member of S_i that appears third in (B.1) as the *third member of* S_i. The *last member* or *bottom member* is the second member if S_i is of size 2, and is the third member is S_i is of size 3. Note that the first, second, and (if it is there) third member of S_i are consecutive in (B.1). The far left column in Fig. 1 shows the link pattern for I_{16}. To the right of each link is its type. For example, $L(0,1)$ has type $(+,+,-)$. The column to the right of the column of link types in Fig. 1) shows the link pattern for I'_{12}. To the left of each link is its type. For example, $L(0,1')$ has type $(+,+,-)$. Later we shall explain the meanings of A_1, A_2, High, Middle, and Low, and of the last two columns of links.

Note that for each S_i of size 2, either (1) the first member ℓ_1 of S_i has type $(+,-,-)$ and the second member ℓ_2 has type $(-,+,+)$, or (2) ℓ_1 has type $(-,-,+)$ and ℓ_2 has type $(+,+,-)$. Hence, in both cases, $\ell_1 \geq \ell_2$ and $\ell_2 \geq \ell_1$. For the set $S_{\frac{m}{2}+1}$ with three members, the first member ℓ_1 has type $(-,-,+)$, the second member ℓ_2 has type $(+,+,-)$, and the third member ℓ_3 has type $(-,+,+)$. Thus, $\ell_1 \geq \ell_2$, $\ell_2 \geq \ell_1$, $\ell_2 \geq \ell_3$, and $\ell_3 \geq \ell_2$.

The only possible preferences between different S_i's can arise for adjacent S_i's, since only then can we have a link in one be in conflict with a link in the other. And in fact, as we now show, between every adjacent pair of S_i's there is indeed a preference, with the bottom element ℓ_1 of S_i (or only element ℓ_1 of S_i, when $i = 1$) being strictly preferred to the top element ℓ_2 of S_{i+1} (or only element ℓ_2 of S_{i+1}, when $i = m$). That is, in all of these cases, we have $\ell_1 > \ell_2$, which means $\ell_1 \geq \ell_2$ but not $\ell_2 \geq \ell_1$. This is because in all of these cases, either (1) ℓ_1 is of type $(+,+,-)$ and ℓ_2 is of type $(+,-,-)$, or (2) ℓ_1 is of type $(-,+,+)$ and ℓ_2 is of type $(-,-,+)$. See Fig. 1.

From the transitive closure of $\geq$, we obtain exactly those strict inequalities $>$ that can be inferred by transitivity of $>$ from the following, where $S_j > S_{j+1}$ means that for each ℓ_1 in S_j and each ℓ_2 in S_{j+1}, we have $\ell_1 > \ell_2$:

$$S_1 > S_2 > \cdots S_m > S_{m+1} \tag{B.3}$$

Similarly, we define the set S'_1 to consist of the link $L(0,1')$, and we define the sets S'_j for $2 \leq j \leq n+1$ just as we defined S_j, except that we replace m by n, and we replace positive integers k in the links with k'. Then just as before, we have:

$$S'_1 > S'_2 > \cdots S'_n > S'_{n+1} \tag{B.4}$$

There is no preference relationship between a member of an S_i and a member of an S'_j.

If $S(a,b)$ is a fact, define $U(S(a,b))$ to be the fact $S(\underline{b},\underline{a})$, where $\underline{a}$ and $\underline{b}$ are new values ("U" stands for "underline"). If S is a binary relation, define $U(S)$ to be the relation obtained by replacing every fact $S(a,b)$ by $U(S(a,b))$. If I is an instance (a set of relations) where each relation is binary, define $U(I)$ to be the result of replacing every relation S of I by $U(S)$. The instance we shall focus on in this proof is $I_{n+4,n} \cup U(I_{n+4,n})$. The links of $U(I_{16,12})$ are shown in the last two columns of Fig. 1.

B.4. Finding the certain links

Recall that the weight of a link ℓ is defined recursively to be $\mathrm{Val}(\ell)$ plus the sum of the weights of all of the links ℓ' such that $\ell > \ell'$. Since $\mathrm{Val}(\ell) = 1$ for $\mathcal{E}$, the weight of a link ℓ is 1 plus the sum of the weights of all of the links ℓ' such that $\ell > \ell'$.

Let D be the domain of $I_{n+4,n}$. Thus, $D = \{0,\ldots,2n+9,1',\ldots(2n+1)'\}$. Let D' be the domain of $U(I_{n+4,n})$. By construction, D and D' are disjoint. Let N be a maximum weight repair (with respect to Σ and $w_{\Sigma,\Pi}$ on $\langle I, I^*\rangle$, where $I = I_{n+4,n} \cup U(I_{n+4,n})$. We shall now investigate the links $L(a,b)$ of N. Since a and b are both in D or both in D', and since D and D' are disjoint, it follows that the presence or absence in N of links derived from $I_{n+4,n}$ have no effect on the presence or absence in N of links derived from $U(I_{n+4,n})$, and vice versa. Therefore, N is of the form $M \cup M'$, where M is a maximum weight repair with respect to Σ and $w_{\Sigma,\Pi}$ on $\langle I, I^*\rangle$, where $I = I_{n+4,n}$, and where M' is a maximum weight repair with respect to Σ and $w_{\Sigma,\Pi}$ on $\langle I, I^*\rangle$, where $I = U(I_{n+4,n})$. Therefore, if X_n is the set of certain links for $I_{n+4,n}$ (and so, by symmetry, $U(X_n)$ is the set of certain links for $U(I_{n+4,n})$), we know that the set of certain links for $I_{n+4,n} \cup U(I_{n+4,n})$ is $X_n \cup U(X_n)$. Therefore, we now determine X_n.

We first show that $L(0,1)$ is a certain link for $I_{n+4,n}$ Assume not; we shall derive a contradiction. Let M be a maximum weight repair that does not contain $L(0,1)$. For each i, let s_i be the weight of each member of S_i (all members of S_i have the same weight). For example,

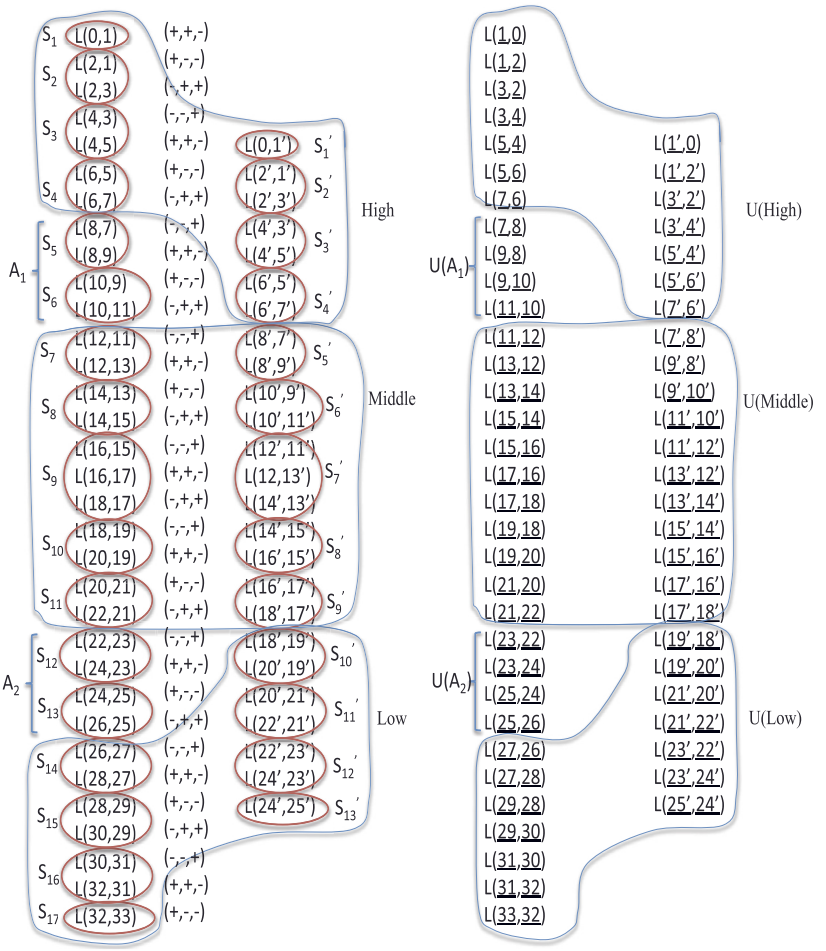

Fig. 1. The link structure for $I_{16,12} \cup U(I_{16,12})$.

$$s_1 = 1 + 2s_2 + \cdots + 2s_{\frac{n}{2}+2} + 3s_{\frac{n}{2}+3} + 2s_{\frac{n}{2}+4} + \cdots 2s_{n+4} + s_{n+5}.$$

The term $2s_2$ comes from the fact that S_2 contains 2 members, both dominated by the member $L(0,1)$ of S_1; the term $3s_{\frac{n}{2}+3}$ comes from the fact that $S_{\frac{n}{2}+3}$ contains 3 members, all dominated by $L(0,1)$; and so on. Similarly, for each j, let s'_j be the weight of each member of S'_i. Now M cannot contain two members of the same S_i if S_i is of size 2, or two members of the same S'_j if S'_j is of size 2, because this would violate an FD. For the same reason, for the sets $S_{\frac{n}{2}+3}$ and $S'_{\frac{n}{2}+1}$ of size 3, we know that M can contain at most 2 members of each.

Let $A = s_2 + s_3 + \cdots + s_{n+5} + s'_1 + \cdots + s'_{n+1}$. From what we have said, and since also M does not contain the only member of S_1, it follows that the weight of M is at most $A + s_{\frac{n}{2}+3} + s'_{\frac{n}{2}+1}$ (we are adding in $s_{\frac{n}{2}+3} + s'_{\frac{n}{2}+1}$ since M could possibly have two members in $S_{\frac{n}{2}+3}$ and two members in $S'_{\frac{n}{2}+1}$). Now $s'_j < s_{j+2}$ for $1 \le j \le n+1$, since, intuitively, we add from the bottom up, and since for each j with $1 \le j \le n+1$, the size of S'_j is at most the size of S_{j+2} (in particular, the set S'_j of size 3 occurs when $j = \frac{n}{2}+1$, and the set S_j of size 3 occurs when $j = \frac{n}{2}+3$). Therefore A is less than

$$s_2 + 2s_3 + 2s_4 + \cdots + 2s_{n+3} + s_{n+4} + s_{n+5} \tag{B.5}$$

Now $s_2 > 2s_{\frac{n}{2}+3} > s_{\frac{n}{2}+3} + s'_{\frac{n}{2}+1}$, where the first inequality follows from the construction of the weight of the S_i's, and the second inequality follows from the fact shown earlier that $s'_j < s_{j+2}$ for $1 \le j \le n+1$. Therefore, the weight of M, which we noted is less than $A + s_{\frac{n}{2}+3} + s'_{\frac{n}{2}+1}$, is less than the weight in (B.5) plus s_2, that is,

$$2s_2 + 2s_3 + 2s_4 + \cdots + 2s_{n+3} + s_{n+4} + s_{n+5} \tag{B.6}$$

But (B.6) is less than the weight of the link $L(0,1)$ by itself, which contradicts the assumption that M is a maximum weight repair. So indeed, $L(0,1)$ is a certain link.

Please cite this article in press as: D. Burdick et al., Expressive power of entity-linking frameworks, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.09.001

We now show that every maximum weight repair M for $I_{n+4,n}$ contains some member of each S_i except possibly the last one (S_{n+5}). Later we shall show that M contains also the (unique) member of S_{n+5}. We already showed that M contains the (unique) member $L(0,1)$ of S_1. Assume that M contains no member of S_i for some i with $2 \le i \le n+4$. In particular, M does not contain the bottom member of S_i. This must be prevented by a conflict with the top member of S_{i+1} (since the conflict cannot be with the other member of S_i, because by assumption S_i contains no member of M). Form M' by removing the top member of S_{i+1} and replacing it by the bottom member of S_i. Then M' is a repair, and it has strictly higher weight than M, since M' was obtained from M by replacing a member of S_{i+1} by a member of S_i. This is our desired contradiction. Therefore, M contains a member of every S_i, except possibly S_{n+5}.

Now M cannot contain the top member of S_2, since that conflicts with the only member $L(0,1)$ of S_1. So M must contain the second member of S_2. Therefore, M cannot contain the top member of S_3, and so it must contain the bottom member of S_3. This process continues, to show us that M must contain the bottom member of $S_{\frac{n}{2}+2}$. So M cannot contain the top member of the 3-element set $S_{\frac{n}{2}+3}$. Therefore, it must contain the second or the third member. If it were to contain the third member, then M would have to contain the bottom member of $S_{\frac{n}{2}+4}$, and so on down the line to containing the bottom member of S_{n+4}. So it could not contain the only member of S_{n+5}. But if M were to contain the second member of the 3-element set $S_{\frac{n}{2}+3}$, then it could contain a member of every S_i. Therefore, each maximum weight repair must contain the second member of the 3-element set $S_{\frac{n}{2}+3}$, the top member of every succeeding 2-element S_i, and the only member of S_{n+5}.

Since M was taken to be an arbitrary maximum weight repair for $I_{n+4,n}$ we now know that $L(0,1), L(2,3), \ldots, L(2n+8, 2n+9)$ are certain links. Let us now see what the certain links are among $S'_1, \ldots, S'_{n+1}$.

No maximum weight repair M can contain the only member $L(0,1')$ of S'_1, since it conflicts with the certain link $L(0,1)$. By an argument like that above, we see that every S'_i, except S'_1 (which, as we have said, is forbidden) and possibly S'_{n+1}, must contain a member of M. If a maximum weight repair were to contain the (only) member of S'_{n+1}, then it would need to contain the top member of S'_i for $\frac{n}{2}+2 \le i \le n$. But then it could not contain the third member of $S'_{\frac{n}{2}+1}$, and so could contain only one member of $S'_{\frac{n}{2}+1}$, along with one member of every remaining S'_i except for S'_1 (which is forbidden). Since the weight of a link is higher in $S'_{\frac{n}{2}+1}$ than in S'_{n+1}, the highest weight we could possibly attain is by having the first and third links in $S'_{\frac{n}{2}+1}$ and one member of every remaining S'_i except for S'_1 and S'_{n+1}. There is exactly one way to attain this, by taking $L(2',1'), L(4',3'), \ldots, L((2n)', (2n-1)')$.

Let S be

$$\left\{L(0,1), L(2,3), \ldots, L(2n+8, 2n+9), L(2',1'), L(4',3'), \ldots, L((2n)', (2n-1)')\right\}.$$

We showed that every member of S is a certain link of $I_{n+4,n}$ No other link can be a certain link for $I_{n+4,n}$, since any other link (which is necessarily of the form $L(a,b)$ where $R(a,b)$ holds) would conflict with a member of S. Therefore, S is the set of certain links for $I_{n+4,n}$. As before, we denote this set of certain links by X_n. As noted earlier, it follows that the set of certain links for $I_{n+4,n} \cup U(I_{n+4,n})$ is $X_n \cup U(X_n)$.

B.5. The strength of allowing preferences

We now show that there is no entity-linking specification $\mathcal{E}'$ in the framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$ of maximum-value solutions that is certain-link equivalent to $\mathcal{E}$. Assume that there were; we shall derive a contradiction. We consider two choices for I, where in both cases we take $J = I^*$. Our first choice is $I = \{R(0,1)\}$. Since $L(0,1)$ is a certain link in $\mathcal{E}$ on $\langle I, J\rangle$ when $I = \{R(0,1)\}$ and $J = I^*$, we must have $L(0,1)$ as a certain link in $\mathcal{E}'$ on $\langle I, J\rangle$. From this we see that the inclusion dependencies for $\mathcal{E}'$ are $L[X| \subseteq R[X]$ and $L[Y| \subseteq R[Y]$.

We now consider our main choice of instances. Let r be as in Theorem 6, where the role of $\mathcal{E}$ is played by $\mathcal{E}'$. For convenience, we assume without loss of generality that $r \ge 3$. Let $n = 4r$, so $n \ge 12$. We then take I to be $I_{n+4,n} \cup U(I_{n+4,n})$ and $J = I^*$. We showed that in $\mathcal{E}$, the certain links are $X_n \cup U(X_n)$, and so by certain-link equivalence, this holds also for $\mathcal{E}'$.

To show that both FDs $L: X \to Y$ and $L: Y \to X$ are constraints of $\mathcal{E}'$, we will make use of Theorem 6 (the Locality Theorem). To do so, we first make an observation. If a is an integer, then let $v(a) = a$, and if a is of the form k' where k is an integer, then let $v(a) = k$. Now every fact in $I_{n+4,n}$ is of one of the forms $R(a,b)$, $W_1(a,b)$, $W_2(a,b)$, or $W_3(a,b)$, where $|v(a) - v(b)| = 1$. We therefore have the following simple fact, which we denote by (*):

(*) If c is in $N_r(a,b)$, then either $|v(c) - v(a)| \le r$ or $|v(c) - v(b)| \le r$.

Throughout the rest of this proof, for each source instance I we write simply N_r for N_r^I. We now show that $I_{n+4,n} \restriction N_r(0,1)$ and $I_{n+4,n} \restriction N_r(0,1')$ are isomorphic under the isomorphism f where $f(0) = 0$, and where $f(i) = i'$ and $f(i') = i$ for each positive integer i. The first place in the sequence (B.1) of links (where $m = n+4$) that the pattern $(+,+,-)$, $(+,-,-)$, $(-,+,+)$, $(-,-,+)$ of types fails to repeat is just after the very middle link $L(n+4, n+5)$ (that is, $L(4r+4, 4r+5)$) and the first place in the sequence (B.2) of links that the pattern $(+,+,-)$, $(+,-,-)$, $(-,+,+)$, $(-,-,+)$ of types fails to repeat is just after the very middle link $L(n', (n+1)')$ (that is, $L((4r)', (4r+1)')$). By (*), we see that no entry of these middle

links is in $N_r(0,1)$ or $N_r(0,1')$, so $I_{n+4,n} \restriction N_r(0,1)$ and $I_{n+4,n} \restriction N_r(0,1')$ are indeed isomorphic under the isomorphism f. Hence, by Theorem 6, we know that $L(0,1)$ and $L(0,1')$ have the same weights in $\mathcal{E}'$.

We now show that both FDs $L: X \to Y$ and $L: Y \to X$ are constraints of $\mathcal{E}'$. Assume not; we shall derive a contradiction. Assume first that the FD $L: Y \to X$ is not a constraint of $\mathcal{E}'$. Now $L(0,1')$ is not a certain link for I in $\mathcal{E}'$, since it is not a certain link for I in $\mathcal{E}$. So let M be a maximum weight repair that does not contain $L(0,1')$. Then of course M contains the certain link $L(0,1)$. As we just showed, $L(0,1)$ and $L(0,1')$ have the same weight in $\mathcal{E}'$, and in particular $L(0,1')$ satisfies the matching constraint for $\mathcal{E}'$. Form M' by replacing $L(0,1)$ by $L(0,1')$. Then M' satisfies the inclusion dependencies, the only possible FD $L: X \to Y$, and the matching constraint. Furthermore, M' has the same weight as M, since $L(0,1)$ and $L(0,1')$ have the same weight, and so M' is a maximum weight repair. But this is a contradiction, since M' is a maximum weight repair that does not contain the certain link $L(0,1)$. So the FD $L: Y \to X$ is a constraint of $\mathcal{E}'$. The proof that the FD $L: X \to Y$ is a constraint of $\mathcal{E}'$ is almost the same, except rather than replacing the certain link $L(0,1)$ in a maximum weight repair by $L(0,1')$, we instead replace the certain link $L(\underline{1},\underline{0})$ by $L(\underline{1'},\underline{0})$. So indeed, both FDs $L: X \to Y$ and $L: Y \to X$ are constraints of $\mathcal{E}'$.

There are now two cases, depending on whether or not $L(\underline{(2n+1)'}, (2n+1)')$ satisfies the matching constraint for $\mathcal{E}'$ (this link of course does not satisfy the matching constraint for $\mathcal{E}$, since the tuple $(\underline{(2n+1)'}, (2n+1)')$ is not in the R relation). Consider first the case where $L(\underline{(2n+1)'}, (2n+1)')$ satisfies the matching constraint for $\mathcal{E}'$. Let M be a maximum weight repair for $\mathcal{E}'$. By certain-link equivalence, we know that M contains $X_n \cup U(X_n)$. According to the inclusion dependency for $L[X]$, and the FD $L: X \to Y$, the only possible first entry (x value) for a tuple that is not already in $X_n \cup U(X_n)$ is $\underline{(2n+1)'}$. Similarly, the only possible second entry (y value) for a tuple that is not already in $X_n \cup U(X_n)$ is $(2n+1)'$. So the only possible fact in M other than those in $X_n \cup U(X_n)$ is $L(\underline{(2n+1)'}, (2n+1)')$. Let $M' = X_n \cup U(X_n) \cup \{L(\underline{(2n+1)'}, (2n+1)')\}$. Then M' satisfies the inclusion dependencies, FDs, and matching constraint, and is the unique maximum weight repair (in particular, $M = M'$). Therefore, $L(\underline{(2n+1)'}, (2n+1)')$ is a certain link in $\mathcal{E}'$. But this is a contradiction, since $\mathcal{E}$ and $\mathcal{E}'$ are certain-link equivalent, and $L(\underline{(2n+1)'}, (2n+1)')$ is not a certain link in $\mathcal{E}$.

In the second case (which we consider for the rest of the proof), $L(\underline{(2n+1)'}, (2n+1)')$ does not satisfy the matching constraint for $\mathcal{E}'$, and so from what we have just discussed, it follows that $X_n \cup U(X_n)$ is the unique maximum weight repair for $\mathcal{E}'$.

B.5.1. Defining High$_i$, Middle$_i$, Low$_i$, A_1, and A_2

We now define some sets of links for $I_{n+4,n}$. Define High_1 to consist of the "highest links" $L(0,1)$ and $L(0,1')$; define High_2 to consist of the "second from the highest" links $L(2,1)$ and $L(2',1')$; define High_3 to consist of the "third from the highest" links $L(2,3)$ and $L(2',3')$; and so on. Let c be such that $\text{High}_c = \{L(\frac{n}{2}, \frac{n}{2}+1), L((\frac{n}{2})', (\frac{n}{2}+1)')\}$, that is (since $n = 4r$), $\{L(2r, 2r+1), L((2r)', (2r+1)')\}$. Let $\text{High} = \cup_{j=1}^{c} \text{High}_j$. Note that by construction, for each i both members of High_i have the same type ($(+,+,-)$, etc.).

Define Middle_0 to consist of the very middle links $L(n+4, n+5)$ (that is, $L(4r+4, 4r+5)$) and $L(n', (n+1)')$ (that is, $L((4r)', (4r+1'))$. These are the links that are the middle links of the 3-element sets $S_{\frac{n}{2}+3}$ and $S'_{\frac{n}{2}+1}$. Define Middle_{-1} to consist of the links just before the very middle links, namely $L(n+4, n+3)$ (that is, $L(4r+4, 4+3)$) and $L(n', (n-1)')$ (that is, $L((4r)', (4r-1'))$. Let e_1 be such that $\text{Middle}_{-e_1} = \{L(\frac{n}{2}+6, \frac{n}{2}+5), L((\frac{n}{2}+2)', (\frac{n}{2}+1)')\}$, that is, $\{L(2r+6, 2r-5), L((2r+2)', (2r+1)')\}$. Define Middle_1 to consist of the links just after the very middle links, namely $L(n+6, n+5)$ (that is, $L(4r+6, 4+5)$) and $L(n+2)', (n+1)')$ (that is, $L((4r+2)', (4r+1)')$). Let e_2 be such that $\text{Middle}_{e_2} = \left\{L(\frac{3n}{2}+4, \frac{3n}{2}+3), L((\frac{3n}{2})', (\frac{3n}{2}-1)')\right\}$, that is, $\{L(6r+4, 6r+3), L((6r)', (6r-1)')\}$. Let $\text{Middle} = \cup_{j=-e_1}^{e_2} \text{Middle}_j$. We now show that for each i both members of Middle_i have the same type. This is true for $i = 0$, since both members of Middle_0 have type $(+,+,-)$. It is then true by forward induction for $i > 0$, and by backward induction for $i < 0$.

Define Low_1 to consist of the lowest links $L(2n+8, 2n+9)$ (that is, $L(8r+8, 8r+9)$) and $L((2n)', (2n+1)')$ (that is, $L((8r)', (8r+1)')$). Define Low_2 to consist of the second from the lowest links $L(2n+8, 2n+7)$ (that is, $L(8r+8, 8r+7)$) and $L((2n)', (2n-1)'$ (that is, $L((8r)', (8r-1)')$), and so on. Let d be such that $\text{Low}_d = \left\{L(\frac{3n}{2}+8, \frac{3n}{2}+9), L((\frac{3n}{2})', (\frac{3n}{2}+1)')\right\}$, that is, $\{L(6r+8, 6r+9), L((6r)', (6r+1)')\}$ and let $\text{Low} = \cup_{j=1}^{d} \text{Low}_j$. We now show that for each i both members of Low_i have the same type. This is true for $i = 1$, since both members of Low_1 have type $(+,-,-)$. It is then true by induction for $i > 0$.

By construction, the sets High, Middle, and Low are pairwise disjoint, and every link for I'_n is in exactly one of High, Middle, and Low.

There are eight links in (B.1) that are not in High, Middle, or Low: four of these lie between High and Middle (these are labeled as A_1 in Fig. 1), and four of these lie between Middle and Low (these are labeled as A_2 in Fig. 1).

B.5.2. Neighborhoods avoiding High$_1$, Middle$_0$, and Low$_1$

The last link in (B.1) that is in High is the link $L(2r, 2r+1)$ of High_c. Therefore, by the fact (*) above we know that the r-neighborhood $N_r(2r, 2r+1)$ corresponding to the last link in (B.1) that is in High does not contain either entry of the very first link $L(0,1)$ if $2r-1 > r$, that is, $r > 1$. Identically, the r-neighborhood $N_r((2r)', (2r+1)')$ corresponding to the last link in (B.2) that is in High does not contain either entry of the very first link $L(0,1')$ if $2r-1 > r$, that is, $r > 1$. Since we

are assuming that $r \geq 3$, it follows that no r-neighborhood $N_r(a,b)$ corresponding to a link $L(a,b)$ *not* in High contains any entry of a member of High_1.

The first link in (B.1) that is in Middle is the link $L(2r+6, 2r+5)$ of Middle_{-e_1}. The very middle link in (B.1) is $L(4r+4, 4r+5)$. Therefore, by the fact (*) above, we know that the r-neighborhood $N_r(2r+6, 2r+5)$ corresponding to the first link in (B.1) that is in Middle does not contain either entry of the very middle link $L(4r+4, 4r+5)$ if $(4r+4)-(2r+6) > r$, that is, if $r > 2$. Similarly, the r-neighborhood $N_r(6r+4, 6r+3)$ corresponding to the last link in (B.1) that is in Middle does not contain either entry of the very middle link $L(4r+4, 4r+5)$ if $(6r+3)-(4r+5) > r$, that is, if $r > 2$. Therefore, since we are assuming that $r \geq 3$, we know that for each link $L(a,b)$ in (B.1) that is not in Middle, the r-neighborhood $N_r(a,b)$ corresponding to the link $L(a,b)$ does not contain either member of Middle_0, the set of the very middle links. Since the number of links in Middle that are in (B.2) is the same as the number of links in Middle that are in (B.1), the same phenomenon holds when we consider links in (B.2).; that is, for each link $L(a,b)$ in (B.2) that is not in Middle, the r-neighborhood $N_r(a,b)$ corresponding to $L(a,b)$ does not contain either entry of Middle_0. So no r-neighborhood $N_r(a,b)$ corresponding to a link $L(a,b)$ *not* i in Middle contains any entry of a member of Middle_0.

The first link in (B.1) that is in Low is the link $L(6r+8, 6r+9)$ of Low_d. Therefore, by the fact (*) above, we know that the r-neighborhood $N_r(6r+8, 6r+9)$ corresponding to the first link in (B.1) that is in Low does not contain either entry of the very last link $L(8r+8, 8r+9)$ if $(8r+8)-(6r+9) > r$, that is, $r > 1$. Since the number of links in Low that are in (B.2) is the same as the number of links in Low that are in (B.1), the same phenomenon holds when we consider links in (B.2). So no r-neighborhood $N_r(a,b)$ corresponding to a link $L(a,b)$ *not* i in Low contains any entry of a member of Low_1.

B.5.3. Equality of weights of links within each High_i, within each Middle_i, and within each Low_i

As before, let f be the function where $f(0) = 0$, and where $f(i) = i'$ and $f(i') = i$ for each positive integer i. If $L(a_1, b_1)$ and $L(a_2, b_2)$ are the two links in High_{i_0}, where $1 \leq i_0 \leq c$, then as we noted in Subsection B.5.2, neither $N_r(a_1, b_1)$ nor $N_r(a_2, b_2)$ contains any member of Middle_0, and so by an argument similar to that in Subsection B.5 involving the repeating pattern of types, we see that f is an isomorphism between $I_{n+4,n} \restriction N_r(a_1, b_1)$ and $I_{n+4,n} \restriction N_r(a_2, b_2)$ with $f(a_1) = a_2$ and $f(b_1) = b_2$. By our choice of r, we know from Theorem 6, where the role of $\mathcal{E}$ is played by $\mathcal{E}'$, that the links $L(a_1, b_1)$ and $L(a_2, b_2)$ have the same weight.

Before we consider links in Middle and Low, we need an intermediate notion. Let us say that a link $L(a,b)$ not in High_1 is *left-matching* if the previous link in the sequence (B.1) or (B.2) is of the form $L(a,x)$ for some x, and *right-matching* if the previous link in the sequence (B.1) or (B.2) is of the form $L(x,b)$ for some x. Thus, a link $L(a, a+1)$ or $L(a', (a+1)')$ is left-matching, and a link $L(a, a-1)$ or $L(a', (a-1)')$ is right-matching. Note that the links alternate: after a right-matching link is a left-matching link, after that is a right-matching link, and so on.

Let $L(a_1, b_1)$ and $L(a_2, b_2)$ be the two links in Middle_{i_0}, where $-e_1 \leq i_0 \leq e_2$. We assume that $L(a_1, b_1)$ is in (B.1) and that $L(a_2, b_2)$ is in (B.2). We now show that $L(a_1, b_1)$ and $L(a_2, b_2)$ have the same weight in $\mathcal{E}'$. Let f be the mapping that maps j to $(j-4)'$ for each j in $N_r(a_1, b_1)$. Then f is an isomorphism between $I_{n+4,n} \restriction N_r(a_1, b_1)$ and $I_{n+4,n} \restriction N_r(a_2, b_2)$ with $f(a_1) = a_2$ and $f(b_1) = b_2$, because (i) $L(a_1, b_1)$ is left-matching if and only if $L(a_2, b_2)$ is left-matching, (ii) from what we have shown, neither $N_r(a_1, b_1)$ nor $N_r(a_2, b_2)$ contains an entry of a link in High_1 (the very highest links) or any entry of a link in Low_1 (the very lowest links), (iii) as we noted in Subsection B.5.1, the members of Middle_{i_0} have the same type, and (iv) in both neighborhoods $N_r(a_1, b_)$ and $N_r(a_2, b_2)$ there is the same pattern of repeating types of links. So again, by our choice of r, we know that the links $L(a_1, b_1)$ and $L(a_2, b_2)$ have the same weight.

Now let $L(a_1, b_1)$ and $L(a_2, b_2)$ be the two links in Low_{i_0}, where $1 \leq i_0 \leq d$. We assume that $L(a_1, b_1)$ is in (B.1) and that $L(a_2, b_2)$ is in (B.2). We now show that $L(a_1, b_1)$ and $L(a_2, b_2)$ have the same weight in $\mathcal{E}'$. Let f be the function that maps j to $(j-8)'$ for each j in $N_r(a_1, b_1)$. Thus, f associates the last link $L(2n+8, 2n+9)$ in (B.1) and the last link $L((2n)', (2n+1)')$ in (B.2). Again, f is an isomorphism between $I_{n+4,n} \restriction N_r(a_1, b_1)$ and $I_{n+4,n} \restriction N_r(a_2, b_2)$ with $f(a_1) = a_2$ and $f(b_1) = b_2$, because (i) $L(a_1, b_1)$ is left-matching if and only if $L(a_2, b_2)$ is left-matching, (ii) from what we have shown, neither $N_r(a_1, b_1)$ nor $N_r(a_2, b_2)$ contains an entry of a link in Middle_0, (iii) as we noted, the members of Low_{i_0} have the same type, and (iv) in both neighborhoods $N_r(a_1, b_)$ and $N_r(a_2, b_2)$ there is the same pattern of repeating types of links. So again, by our choice of r, we know that the links $L(a_1, b_1)$ and $L(a_2, b_2)$ have the same weight.

B.5.4. Contradiction via showing there is another maximum weight repair

Let $M = X_n \cup U(X_n)$. Let M' consist of all of the links $L(a,b)$ where $R(a,b)$ is a fact of $I_{n+4,n} \cup U(I_{n+4,n})$ but $L(a,b)$ is not in $X_n \cup U(X_n)$. Thus, M' is the union of

$$L(2,1), L(4,3), \ldots, L(2n+8, 2n+7), L(0,1'), L(2',3'), \ldots, L((2n)', (2n+1)')$$

and

$$L(\underline{1},\underline{2}), L(\underline{3},\underline{4}), \ldots, L(\underline{2n+7}, \underline{2n+8}), L(\underline{1'},\underline{0}), L(\underline{3'},\underline{2'}), \ldots, L(\underline{(2n+1)'}, \underline{(2n)'}).$$

Note that both M and M' have $4n+10$ links.

We already noted that M is the unique maximum weight repair. We shall show shortly that M' is also a maximum weight repair, which gives us our desired contradiction.

Please cite this article in press as: D. Burdick et al., Expressive power of entity-linking frameworks, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.09.001

We now show that every member of each High_i has one member of M and one member of M'. This is certainly true of High_1, which contains $L(0,1)$, which is in M, and $L(0,1')$, which is in M'. Since M contains every other member of the sequence (B.1) starting with $L(0,1)$, and since M contains every other member of the sequence (B.2) starting with $L(2',1')$, we see inductively that indeed, every member of each High_i has one member of M and one member of M'.

We now show that every member of each Middle_i has one member of M and one member of M'. This is certainly true of Middle_0, which contains $L(n',(n+1)')$, which is in M, and $L(n+4,n+5)$, which is in M'. Again by induction (going forward in the sequence (B.1) starting with $L(n+4,n+5)$, and forward in the sequence (B.2), starting with $L(n',(n+1)')$), and by reverse induction (going backward in the sequence (B.1) starting with $L(n+4,n+5)$, and backward in the sequence (B.2) starting with $L(n',(n+1)')$), we see that indeed, every member of each Middle_i has one member of M and one member of M'.

We now show that every member of each Low_i has one member of M and one member of M'. This is certainly true of Low_1, which contains $L(2n+8,2n+9)$, which is in M, and $L((2n)',(2n+1)')$ which is in M'. By backward induction (going backward in the sequence (B.1), starting with $L(2n+8,2n+9)$, and backward in the sequence (B.2), starting with $L((2n)',(2n+1)')$), we see that indeed, every member of each Low_i has one member of M and one member of M'.

We just showed that each High_i, each Middle_i, and each Low_i consists of a member of M and a member of M'. We also showed earlier that both members of each High_i have the same weight, both members of each Middle_i have the same weight, and both members of each Low_i have the same weight. Therefore, the sum of the weights of the links in M that are in $\text{High} \cup \text{Middle} \cup \text{Low}$ equals the sum of the weights of the links in M' that are in $\text{High} \cup \text{Middle} \cup \text{Low}$. By symmetry, the sum of the weights of the links in M that are in $U(\text{High}) \cup U(\text{Middle}) \cup U(\text{Low})$ equals the sum of the weights of the links in M' that are in $U(\text{High}) \cup U(\text{Middle}) \cup U(\text{Low})$.

We now consider the remaining links (the links that are not in $\text{High} \cup \text{Middle} \cup \text{Low} \cup U(\text{High}) \cup U(\text{Middle}) \cup U(\text{Low})$). We call these *missing links*. What are they? As we noted earlier, every link for I'_n is in exactly one of High, Middle, and Low, and there are 8 links in (B.1) that are not in High, Middle, or Low: 4 missing links that are consecutive links in the top portion of (B.1), and 4 missing links that are consecutive links in the bottom portion of (B.1.) These 4 missing links in the top portion form the set

$$A_1 = \left\{ L(\frac{n}{2}+2, \frac{n}{2}+1), L(\frac{n}{2}+2, \frac{n}{2}+3), L(\frac{n}{2}+4, \frac{n}{2}+3), L(\frac{n}{2}+4, \frac{n}{2}+5) \right\},$$

and the 4 missing links in the bottom portion form the set

$$A_2 = \left\{ L(\frac{3n}{2}+4, \frac{3n}{2}+5), L(\frac{3n}{2}+6, \frac{3n}{2}+5), L(\frac{3n}{2}+6, \frac{3n}{2}+7), L(\frac{3n}{2}+8, \frac{3n}{2}+7) \right\}.$$

Again, see Fig. 1. Similarly, the links in $U(A_1)$ and $U(A_2)$ are missing links (the only remaining missing links). So the links we have not studied (the links that are not in $\text{High} \cup \text{Middle} \cup \text{Low} \cup U(\text{High}) \cup U(\text{Middle}) \cup U(\text{Low})$) are the 4 links in A_1, the 4 links in A_2, the 4 links in $U(A_1)$, and the 4 links in $U(A_2)$.

For ease in description, let us refer to links of type $(+,+,-)$ as being of *Type 1*, links of type $(+,-,-)$ as being of *Type 2*, links of type $(-,+,+)$ as being of *Type 3*, and links of type $(-,-,+)$ as being of *Type 4*. Of course, a link $L(\underline{b},\underline{a}))$ in $U(A_1) \cup U(A_2)$ has the same type as the link $L(a,b)$ in $A_1 \cup A_2$, and $L(\underline{b},\underline{a}))$ is in M if and only if $L(a,b)$ is in M. Note that the links in $A_1 \cap M$ are of types 1 and 3, but (and this is the reason that we skipped a type in the middle) the links in $A_2 \cap M$ are of types 2 and 4. Since the links in A_1 are consecutive, A_1 has exactly one link of each of the 4 types, and similarly for A_2. Since the links in $A_1 \cap M$ are of types 1 and 3, it follows that the links in A_1 of types 1 and 3 are in M, and the links in A_1 of types 2 and 4 are in M'. Similarly, since the links in $A_2 \cap M$ are of types 2 and 4, it follows that the links in A_2 of types 1 and 3 are in M', and the links in A_2 of types 2 and 4 are in M.

We extend the notion of *left-matching* and *right-matching* to $U(A_1) \cup U(A_2)$ in the natural way: a link $L(\underline{b},\underline{a})$ in $U(A_1) \cup U(A_2)$ is left-matching if the previous link is of the form $L(\underline{b},x)$ for some x, and right-matching if the previous link is of the form $L(x,\underline{a})$ for some x. Again, the links alternate between left-matching and right-matching. Note that the link $L(\underline{b},\underline{a})$ of $U(A_1) \cup U(A_2)$ is left-matching if and only if the link $L(a,b)$ of $A_1 \cup A_2$ is right-matching.

The link of type 1 in A_1, namely the link $L(\frac{n}{2}+2, \frac{n}{2}+3)$, is left-matching. For ease in notation, let us denote this link by $L(a_1,b_1)$. Because of the skip in link types just after the very middle link, the link in A_2 of type 1, namely the link $L(\frac{3n}{2}+6, \frac{3n}{2}+5)$, is right-matching. For ease in notation, let us denote this link by $L(a_2,b_2)$. So the link $L(\underline{b_2},\underline{a_2})$ of type 1 in $U(A_2)$ is left-matching. We just showed that the link in A_1 of type 1 is left-matching, and the link in $U(A_2)$ of type 1 is also left-matching (in fact, this is why we introduced $U(I_{n+4,n})$). From what we showed earlier, we know that neither $N_r(a_1,b_1)$ nor $N_r(\underline{b_2},\underline{a_2})$ contains any entry of a member of High_1, Middle_0, or Low_1. It follows that there is an isomorphism f between $(I_{n+4,n} \cup U(I_{n+4,n})) \restriction N_r(a_1,b_)$ and $(I_{n+4,n} \cup U(I_{n+4,n})) \restriction N_r(\underline{b_2},\underline{a_2})$ with $f(a_1) = \underline{b_2}$ and $f(b_1) = \underline{a_2}$, because in both neighborhoods $N_r(a_1,b_)$ and $N_r(\underline{b_2},\underline{a_2})$ there is the same pattern of repeating types of links. So by Theorem 6, the weight of the type 1 link in A_1 equals the weight of the type 1 link in $U(A_2)$. By the same argument, it follows that the weight of the type i link in A_1 equals the weight of the type i link in $U(A_2)$, for $1 \le i \le 4$. When we combine this with the fact that the link of type i in A_1 is in M if and only if the link of type i in $U(A_2)$ is in M', we see that the sum of the weights of the links in M that are in $A_1 \cup U(A_2)$ equals the sum of the weights of the links in M' that are in $A_1 \cup U(A_2)$. By symmetry, the sum of the weights of the links in M that are in $U(A_1) \cup A_2$ equals the sum of the weights of the links in M' that are in $U(A_1) \cup A_2$.

Please cite this article in press as: D. Burdick et al., Expressive power of entity-linking frameworks, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.09.001

When we combine this with the facts we have shown that the sum of the weights of the links in M that are in $\text{High} \cup \text{Middle} \cup \text{Low}$ equals the sum of the weights of the links in M' that are in $\text{High} \cup \text{Middle} \cup \text{Low}$, and the fact that the sum of the weights of the links in M that are in $U(\text{High}) \cup U(\text{Middle}) \cup U(\text{Low})$ equals the sum of the weights of the links in M' that are in $U(\text{High}) \cup U(\text{Middle}) \cup U(\text{Low})$, we see that the total sum of the weights of the links in M equals the total sum of the weights of the links in M'. Now M' is a repair, since (i) from what we have shown, every link in M' has the same weight as a link in M, and so the matching constraint is satisfied for the links in M', (ii) M' satisfies the inclusion dependencies, and (iii) M' does not violate either FD. But then M' is a maximum weight repair, which is a contradiction, since we showed that M is the unique maximum weight repair. This contradiction shows that $\mathcal{E}'$ does not exist. □

References

[1] Arvind Arasu, Christopher Re, Dan Suciu, Large-scale deduplication with constraints using Dedupalog, in: ICDE, 2009, pp. 952–963.
[2] Marcelo Arenas, Leopoldo E. Bertossi, Jan Chomicki, Consistent query answers in inconsistent databases, in: PODS, 1999, pp. 68–79.
[3] Stephen H. Bach, Hinge-Loss Markov Random Fields and Probabilistic Soft Logic: A Scalable Approach to Structured Prediction, PhD thesis, University of Maryland, 2015.
[4] Stephen H. Bach, Matthias Broecheler, Bert Huang, Lise Getoor, Hinge-loss Markov random fields and probabilistic soft logic, CoRR, http://arxiv.org/abs/1505.04406, 2015.
[5] Zeinab Bahmani, Leopoldo E. Bertossi, Nikolaos Vasiloglou, ERBlox: combining matching dependencies with machine learning for entity resolution, Int. J. Approx. Reason. 83 (2017) 118–141.
[6] Leopoldo E. Bertossi, Solmaz Kolahi, Laks V.S. Lakshmanan, Data cleaning and query answering with matching dependencies and matching functions, Theory Comput. Syst. 52 (3) (2013) 441–482.
[7] Indrajit Bhattacharya, Lise Getoor, Collective entity resolution in relational data, ACM Trans. Knowl. Discov. Data 1 (1) (2007).
[8] Matthias Bröcheler, Lilyana Mihalkova, Lise Getoor, Probabilistic similarity logic, in: UAI 2010, Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, July 8–11, 2010, 2010, pp. 73–82.
[9] Douglas Burdick, Ronald Fagin, Phokion G. Kolaitis, Lucian Popa, Wang-Chiew Tan, A declarative framework for linking entities, ACM Trans. Database Syst. 41 (3) (2016) 17. Preliminary version appeared in ICDT, 2015, pp. 25–43.
[10] Douglas Burdick, Ronald Fagin, Phokion G. Kolaitis, Lucian Popa, Wang Chiew Tan, Expressive power of entity-linking frameworks, in: 20th International Conference on Database Theory, ICDT 2017, March 21–24, 2017, Venice, Italy, 2017, 10.
[11] Jan Chomicki, Jerzy Marcinkowski, Minimal-change integrity maintenance using tuple deletions, Inf. Comput. 197 (1–2) (2005) 90–121.
[12] Xin Dong, Alon Y. Halevy, Jayant Madhavan, Reference reconciliation in complex information spaces, in: SIGMOD, 2005, pp. 85–96.
[13] Jianfeng Du, Guilin Qi, Yi-Dong Shen, Weight-based consistent query answering over inconsistent SHIQ knowledge bases, Knowl. Inf. Syst. 34 (2) (2013) 335–371.
[14] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, Vassilios S. Verykios, Duplicate record detection: a survey, IEEE Trans. Knowl. Data Eng. 19 (1) (2007) 1–16.
[15] Wenfei Fan, Dependencies revisited for improving data quality, in: PODS, 2008, pp. 159–170.
[16] Ivan P. Fellegi, Alan B. Sunter, A theory for record linkage, J. Am. Stat. Assoc. 64 (328) (1969) 1183–1210.
[17] Haim Gaifman, On local and non-local properties, in: Proc. Herbrand Symp. – Logic Colloquium '81, 1982.
[18] Helena Galhardas, Daniela Florescu, Dennis Shasha, Eric Simon, Cristian-Augustin Saita, Declarative data cleaning: language, model, and algorithms, in: VLDB, 2001, pp. 371–380.
[19] Mauricio A. Hernández, Georgia Koutrika, Rajasekar Krishnamurthy, Lucian Popa, Ryan Wisnesky, HIL: a high-level scripting language for entity integration, in: EDBT, 2013, pp. 549–560.
[20] Mauricio A. Hernández, Salvatore J. Stolfo, The merge/purge problem for large databases, in: SIGMOD, 1995, pp. 127–138.
[21] Hanna Köpcke, Erhard Rahm, Frameworks for entity matching: a comparison, Data Knowl. Eng. 69 (2) (2010) 197–210.
[22] Hanna Köpcke, Andreas Thor, Erhard Rahm, Evaluation of entity resolution approaches on real-world match problems, Proc. VLDB Endow. 3 (1) (2010) 484–493.
[23] Nick Koudas, Sunita Sarawagi, Divesh Srivastava, Record linkage: similarity measures and algorithms, in: SIGMOD, 2006, pp. 802–803.
[24] Leonid Libkin, Logics with counting and local properties, ACM Trans. Comput. Log. 1 (1) (2000) 33–59.
[25] Leonid Libkin, Elements of Finite Model Theory, Texts in Theoretical Computer Science. An EATCS Series, Springer, 2004.
[26] Andrei Lopatenko, Leopoldo E. Bertossi, Complexity of consistent query answering in databases under cardinality-based and incremental repair semantics, in: ICDT, 2007, pp. 179–193.
[27] Matthew Richardson, Pedro Domingos, Markov logic networks, Mach. Learn. 62 (1–2) (2006) 107–136.
[28] Slawek Staworko, Jan Chomicki, Jerzy Marcinkowski, Prioritized repairing and consistent query answering in relational databases, Ann. Math. Artif. Intell. 64 (2–3) (2012) 209–246.

Please cite this article in press as: D. Burdick et al., Expressive power of entity-linking frameworks, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.09.001