tension of $\Delta$ to arbitrary concept descriptions is defined via the constructors in Figure 2.

Reference to concrete data objects is accomplished via the parameterization of $\mathcal{EL}^{++}$ by concrete domains $D_1, \ldots, D_n$, which correspond to OWL data types. Such concrete domains are pairs $(\Delta^D, \mathcal{P}^D)$, where $\Delta^D$ is a set and $\mathcal{P}^D$ is a set of predicate names; each $p \in \mathcal{P}^D$ has an arity $n > 0$ and an extension $p^D \subseteq (\Delta^D)^n$. The link between the DL and concrete domains is accomplished via the introduction of feature names $F$ and the concrete domain constructor included in Figure 2; $p$ is used to denote a predicate of a concrete domain, and $f_1, \ldots, f_k$ to denote feature names. The interpretation function maps each feature name $f$ to a partial function from $\Delta^2$ to $\bigcup_{1 \leq i \leq n} \Delta^{D_i}$, where in general it is assumed that $\Delta^{D_i} \cap \Delta^{D_j} = \emptyset$ for $1 \leq i < j \leq n$.

$\mathcal{EL}^{++}$ Knowledge Bases. A KB consists of two sets, referred to as the ABox and the TBox, respectively containing the extensional knowledge about individual objects and intensional knowledge about the general notions for the domain in question. The ABox formally consists of a finite set of concept assertions and role assertions, while the TBox is comprised of a finite set of constraints, which can be general concept inclusions (GCIs), role inclusions (RIs), domain restrictions (DRs), or range restrictions (RRs) (cf. Figure 2). An interpretation $I$ is a model of a TBox $\mathcal{T}$ (resp., ABox $A$) iff, for each contained constraint (resp., assertion), the conditions in the “Semantics” column of Figure 2 are satisfied. We note that the expressive power of $\mathcal{EL}^{++}$ allows the expression of role hierarchies, role equivalences, transitive roles, reflexive roles, left- and right-identity rules, disjointness of complex concept descriptions, and the identity and distinctness of individuals.

Finally, here we adopt the syntactic restriction presented in [1] to avoid intractability/undecidability, which prevents intricate interplay between role inclusions and range restrictions; we do not describe it here for reasons of space.

2.2 Markov Logic Networks

Markov logic networks (MLNs) [20] combine first-order logic with Markov networks (abbreviated MNs; they are also known as Markov random fields) [19]. We now provide a brief introduction first to MNs, and then to MLNs.

Markov Networks. A Markov network (MN) is a probabilistic model that represents a joint probability distribution over a (finite) set of random variables $X = \{X_1, \ldots, X_n\}$. Each random variable $X_i$ may take on values from a finite domain $\text{Dom}(X_i)$. A value for $X = \{X_1, \ldots, X_n\}$ is a mapping $x: X \rightarrow \bigcup_{i=1}^n \text{Dom}(X_i)$ such that $x(X_i)$ is in $\text{Dom}(X_i)$; the domain of $X_i$, denoted $\text{Dom}(X_i)$, is the set of all values for $X$. An MN is similar to a Bayesian network (BN) in that it includes a graph $G = (V, E)$ in which each node corresponds to a variable, but, differently from a BN, the graph is undirected; in an MN, two variables are connected by an edge in $G$ iff they are conditionally dependent. Furthermore, the model contains a potential function $\phi_i$ for each (maximal) clique in the graph; potential functions are non-negative real-valued functions of the values of the variables in each clique (called the state of the clique). In this work, we assume the log-linear representation of MNs, which involves defining a set of features of such states; a feature is a real-valued function of the state of a clique (we only consider binary features in this work). Given a value $x \in \text{Dom}(X)$ and a feature $f_j$ for clique $j$, the probability distribution represented by an MN is given by $P(X = x) = \frac{1}{Z} \exp \left( \sum_{i,j} w_j \cdot f_j(x) \right)$, where $j$ ranges over the set of cliques in the graph $G$, and $w_j = \log \phi_i(x_{(j)})$ (here, $x_{(j)}$ is the state of the $j$-th clique). The term $Z$ is a normalization constant to ensure that the values given by the equation above are in $[0, 1]$; it is given by $Z = \sum_{x \in \text{Dom}(X)} \exp \left( \sum_{j} w_j \cdot f_j(x) \right)$. Probabilistic inference in MNs is tractable; however, approximate inference mechanisms, such as Markov Chain Monte Carlo, have been developed and successfully applied.

Markov Logic Networks. In the following, let $\Delta_{MLN}$, $\mathcal{V}_{MLN}$, and $\mathcal{R}_{MLN}$ denote the set of constants, variables, and predicate symbols. The main idea behind Markov logic networks (MLNs) is to provide a way to soften the constraints imposed by a set of classical logic formulas. Instead of considering worlds that violate some formulas to be impossible, we wish to make them less probable. An MLN is a finite set $L$ of pairs $(F_i, w_i)$, where $F_i$ is a formula in first-order logic over $\mathcal{V}_{MLN}$, $\Delta_{MLN}$, and $\mathcal{R}_{MLN}$, and $w_i$ is a real number. Such a set $L$, along with a finite set of constants $\Delta_{MLN}$, defines a Markov network $M$ that contains: (i) one binary node corresponding to each element of the Herbrand base of the formulas in $L$ (i.e., all possible ground instances of the atoms), where the node’s value is 1 iff the atom is true; and (ii) one feature for every possible