Algorithm anytimeRank(KB = (O, M), stopCond)
// M is assumed to be a ground cMLN
1. score:= empty mapping from atoms to \( \mathbb{R} \) (default 0);
2. out:= empty set of atoms;  \( i := 1 \);
3. while (\( i \leq 2|M| \)) and !stopCond do begin
   // i ranges over classes of possible worlds
4. \( C := \text{compMostProbEqClass}(M, i) \);
5. \( S := \emptyset \);
6. \( i := i + 1 \);
7. while (\( |S| \neq |C| \) ) and !stopCond do
8. \( \lambda := \text{computePossWorld}(C, S) \);
// compute world s.t. \( \lambda \in C \) and \( \lambda \notin S \)
9. \( S := S \cup \{ \lambda \} \);
10. \( O_\lambda := \text{getInducedOnt}(O, \lambda) \);
11. for all atoms \( a \in \text{atomicCons}(O_\lambda) \) do
12. \( \text{score}(a) += \exp \left( \sum_{F_j \in M; C := F_j} w_j \right) \);
13. out:= out \cup \text{atomicCons}(O_\lambda);
14. end;
15. return out sorted in dec. order according to \( \text{score} \).

Figure 4: An anytime algorithm to compute the answer to a subroutines).

<table>
<thead>
<tr>
<th>Class</th>
<th>( f_1 )</th>
<th>( f_2 )</th>
<th>( f_5 )</th>
<th>( f_8 )</th>
<th>( f_9 )</th>
<th>( \sum w_j )</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6.9</td>
</tr>
<tr>
<td>C_2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>6.1</td>
</tr>
<tr>
<td>C_3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>6.1</td>
</tr>
<tr>
<td>C_4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>6.1</td>
</tr>
<tr>
<td>C_5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5.3</td>
</tr>
</tbody>
</table>

Figure 5: Equivalence classes from Example 6, sorted in descending order of the \( \text{score} \) assigned to possible worlds that belong to them (\( e \) to the power of the value in the last column); only 5 of the 64 classes are shown here.

subset of the worlds will be inspected; since the probabilities of worlds in a given equivalence class are all equal (Theorem 4), and this value depends directly on the formulas in the cMLN that are satisfied by the class, the iteration through the equivalence classes in decreasing order of probability is the optimal path to take. Though the total running time of course depends on the stopping condition, Theorem 3, along with the way in which equivalence classes are manipulated (as described above), guarantee a running time that is polynomial in the data complexity as long as the combined number of inspected worlds and equivalence classes is bounded by a polynomial as well.

4.2.2 Bounding the Error of anytimeRank

The following proposition provides a bound on the total “mass” of score that remains unassigned by our algorithm after a certain number of iterations.

**Proposition 2.** Let \( KB = (O, M) \) be a TCP ontology where \( M \) is a ground cMLN with \( n \) ground atoms, and let \( C_1, ..., C_{2^|M|} \) be the set of equivalence classes of \( M \) sorted in decreasing order of their score. Then, after analyzing \( s \) worlds and \( t \) classes with Algorithm anytimeRank, the total unassigned class score mass is bounded by above by

\[
U = \left( 2^n - s \cdot \exp \left( \sum_{F_j \in M; C := F_j} w_j \right) \right).
\]

This result is useful, for instance, in determining a provably correct partial order over the output of the algorithm. For example, if the output is \( \{(a, 120), (b, 90), (c, 80), (d, 10)\} \) and \( U = 30 \), we can safely conclude that \( Pr_{KB}(a) > Pr_{KB}(c) \), \( Pr_{KB}(b) > Pr_{KB}(d) \), and \( Pr_{KB}(d) > Pr_{KB}(b) \).

**Theorem 6.** Let out be the output of Algorithm anytimeRank and \( U \) be the bound on the unassigned score mass as computed in Proposition 2. The partial order \( \leq_U \) defined as: \( a \leq_U b \) iff \( s_a + U \leq s_b \), where \( (a, s_a), (b, s_b) \in \text{out} \), is such that if \( a \leq_U b \) then \( Pr(a) \leq Pr(b) \).

Therefore, Theorem 6 allows us to glimpse into the total order over the set of atoms as established by the true probability values, without actually computing them.

5 Related Work

Ontology languages, rule-based systems, and their integrations are central for the Semantic Web [2]. Although many approaches exist to tight, loose, or hybrid integrations of ontology languages and rule-based systems, to our knowledge there is very little work on the combination of tractable description logics with MLNs. Probabilistic ontology languages in the literature can be classified according to the underlying ontology language, the supported forms of probabilistic knowledge, and the underlying probabilistic semantics (see [14] for a recent survey). Some early approaches [11] generalize the description logic \( \text{ALC} \) and are based on propositional probabilistic logics, while others [12] generalize the tractable DL \( \text{CLASSIC} \) and \( \mathcal{F} \mathcal{L} \), and are based on Bayesian networks as underlying probabilistic semantics. The fairly recent approach in [13], generalizing the expressive DL \( \mathcal{S}\mathcal{H}\mathcal{I}\mathcal{F}(\mathcal{D}) \) and \( \mathcal{S}\mathcal{H}\mathcal{O}\mathcal{L}(\mathcal{D}) \) behind the sublanguages \text{OWL Lite} and \text{OWL DL}, respectively, of the Web ontology language \text{OWL} [18], is based on probabilistic default logics, and allows for rich probabilistic terminological and assertional knowledge. Other recent approaches [23] generalize \text{OWL} by probabilistic uncertainty using Bayesian networks.

In the probabilistic description logics literature, the most closely related work is that of Prob-\( \mathcal{EL} \) [15, 10], a probabilistic extension to \( \mathcal{EL} \) that belongs to a family of probabilistic DLs derived in a principled way from Halpern’s probabilistic first-order logic [4]. One limitation in this line of research is that probabilistic annotations are somewhat restricted, leading to the inability to express uncertainty about certain kinds of general knowledge; note that our formalism does not suffer from this drawback, since probabilistic annotations can be associated with any axiom. In [15], the authors study various logics in this family and