1 Introduction

As defined by the Medical Waste Tracking Act of 1988, clinical solid waste consists of solid waste materials which are generated during diagnosis, treatment, vaccination, research or in the production or testing of biological products for humans and animals, including syringes, live vaccines, blood and other waste contaminated with bodily fluids and removed body organs, among others. This class of waste is now recognized as a potencially hazardous agent affecting both the environment and the human being, due to the labor intensive operations entailed in its collection, segregation and disposal, which involve many possibilities of direct contact with the waste and therefore increase the risk of infections for health care workers, the general public and waste handlers in particular [1,2]. Focusing on the latter, poor management practices and improper precautions taken by clinical waste workers during these operations are quoted as being the main reason of the spread of infectious diseases among clinical waste handlers [3,4], which raises the additional need for adequate risk management strategies ensuring assessment, control, review and identification of risk. Nevertheless, several studies [3–6] indicate that the clinical solid waste management at healthcare facilities is still inadequate in developed countries and that, in many situations, this class of waste is handled and disposed together with non-clinical waste. Given this scenario, the development of strategies for the definition of the best appropriate clinical waste management practice towards the minimization of occupational incidents and environmental contamination is of great importance. One of the main concerns still requiring addressing is the lack of awareness of both healthcare and clinical waste workers in what regards waste differential classification [1].

Taking into consideration the urgent need to bridge and amend these gaps, the present study aims at analysing and developing an artificial intelligence-driven approach for dealing with the judgement difficulties that arise from the waste classification process, especially in environments with defective information, based on the approach presented by *Neves et. al* in [7] and selectively focusing on the clinical waste management situation in Portugal.

2 Case Study

In Portugal, the management of clinical solid waste is regulated by law, and its legal constraints are defined within the Portuguese Legislative Decree no. 178/2006 - September 5th [8]. Particularly, this document provides a classification system for clinical solid waste based on four distinct criteria: typology, danger, production site and treatment required [9]; which allows for a clear and objective assortment of waste samples in four main classes (Figure 1), based on their specific combination of criteria.

Class	Class Description
C1	Equivalent to Municipal Solid Waste (MSW).
C2	Non-biohazardous hospital waste.
C3	Biohazardous hospital waste.
C4	Specific hospital waste.

Figure 1: Portuguese classification system for clinical solid waste.

In this study, the required treatment associated with each class was ignored, as it is impractible to determine the most suitable treatment for a waste sample without knowing beforehand the class it falls into, and the remainder criteria were fitted to the specific description of the classes and thus redefined as: source, type and contamination. According to the aforementioned, a database model was constructed (Figure 3), comprising a primary table and three secondary tables which refer to the analysed cases (Figure 2) and each of the considered criteria, respectively.

ID_Waste	Case Description
1	Scalpel contaminated with blood, provenient from the operating room.
2	Food waste collected from the maternity services.
3	Non-contaminated plaster piece from the orthopedic services.
4	Pair of medical gloves used in waste handling, contaminated with blood, organic fluids or both.
5	Package contaminated with cytostatic agents, either provenient from the labs or medical services.
6	Non-contaminated generic package, provenient from an unknown source.

Figure 2: Description of the cases analysed in the study.

2.1 Knowledge Representation and Reasoning

In regular Logic Programming (LP), the negative information is implicit — in other words, it is not possible to explicity state falsity and propositions are assumed false if there is no reason to believe otherwise. However, explicit negative information plays an important role in natural discourse and commonsense reasoning, and so, for use in deductive databases, knowledge representation and non-monotonic reasoning, a second kind of negation is included, giving rise to the Extended Logic Programming (ELP) paradigm [10]. An extended logic program is a finite set of clauses in the form:

$\mathtt{q} \leftarrow \mathtt{q}_1 \wedge \mathtt{p}_n$ $\mathtt{not}$ $\mathtt{q}_1 \wedge \ldots \wedge$ $\mathtt{not}$ $\mathtt{q}_m$

$?\mathtt{p}_1 \wedge \ldots \wedge \mathtt{p}_n \wedge$ $\mathtt{not}$ $\mathtt{q}_1 \wedge \ldots \wedge$ $\mathtt{not}$ $\mathtt{p}_m$ $(\mathtt{n},\mathtt{m} \geqslant 0)$

where ? is a domain denoting falsity, $\mathtt{p}_i$, $\mathtt{q}_j$ and $\mathtt{q}$ represent classical ground literals — either positive atoms or atoms preceeded by the classical negation sign [11]. In this representation formalism, every program is associated with a set of abducibles [12,13], given here in the form of exceptions to the extensions of the predicates that compose the program.

In order to reason about the body of knowledge presented through the analysed cases, the relations defined in the database model were first rewriten in terms of the following predicates:

```
waste: ID_Waste x Class x ID_Source x ID_Type x ID_Contamination

source: ID_Source x GS x SS x WS x HS x LS x MS

type: ID_Type x GP x FW x CP x OM x DP x OM x IP x MW x NB x SW x IB x MD x CH x AC x CO

contamination: ID_Contamination x BL x OF x IA x CA
```

Subsequently, the extension of the predicates was set in the form of four programs.

2.1.1 Extended Logic Programs

Program 1: Extended logic program for the predicate `waste`.

{

$$\neg\ \mathtt{waste\ (ID,\ CL,\ SR,\ TY,\ CT)} \leftarrow$$
$$\mathtt{not(waste\ (ID,\ CL,\ SR,\ TY,\ CT)),}$$
$$\mathtt{not(abducible}_{waste}\ \mathtt{(ID,\ CL,\ SR,\ TY,\ CT)).}$$

waste(1,C,6,1,1).

waste(2,C,4,2,5).

waste(3,C,6,4,5).

$$\text{waste}(4,C,3,6,\{2,3\}).$$
$$\text{waste}(5,C,\{5,6\},3,4).$$
$$\text{waste}(6,C,\perp,1,5).$$

$\texttt{abducible}_{waste}$ (4, C, 3, 6, 2).
$\texttt{abducible}_{waste}$ (4, C, 3, 6, 3).
$\texttt{abducible}_{waste}$ (5, C, 5, 3, 4).
$\texttt{abducible}_{waste}$ (5, C, 6, 3, 4).
?(($\texttt{abducible}_{waste}$ (ID, CL, SR_1, TY, CT)
$\vee$
$\texttt{abducible}_{waste}$ (ID, CL, SR_2, TY, CT))
$\wedge$
$\neg$ ($\texttt{abducible}_{waste}$ (ID, CL, SR_1, TY, CT)
$\wedge$
$\texttt{abducible}_{waste}$ (ID, CL, SR_2, TY, CT)))
$\texttt{abducible}_{waste}$ (6, C, $\perp$, 1, 5).
$\texttt{abducible}_{waste}$ (ID, CL, SR, TY, CT) $\leftarrow$ waste (ID, CL, $\perp$, TY, CT).
}

Program 2: Extended logic program for the predicate source.

{

$\neg$ source (ID,GS,SS,WS,HS,LS,MS) $\leftarrow$
not(source (ID,GS,SS,WS,HS,LS,MS)),
not($\texttt{abducible}_{source}$ (ID,GS,SS,WS,HS,LS,MS)).

$$\text{source}(1,1,0,0,0,0,0).$$
$$\text{source}(2,0,1,0,0,0,0).$$
$$\text{source}(3,0,0,1,0,0,0).$$
$$\text{source}(4,0,0,0,1,0,0).$$
$$\text{source}(5,0,0,0,0,1,0).$$
$$\text{source}(6,0,0,0,0,0,1).$$

}

Program 3: Extended logic program for the predicate type.

{

$\neg$ type (ID, GP, FW, CP, OM, DP, IP, MW, NB, SW, IB, MD, CH, AC, CO) $\leftarrow$
not(type (ID, GP, FW, CP, OM, DP, IP, MW, NB, SW, IB, MD, CH, AC, CO)),

`not(abducible`$_{type}$ `(ID, GP, FW, CP, OM, DP, IP, MW, NB, SW, IB, MD, CH, AC, CO)).`

type(1,1,0,0,0,0,0,0,0,0,0,0,0,0,0).
type(2,0,1,0,0,0,0,0,0,0,0,0,0,0,0).
type(3,0,0,1,0,0,0,0,0,0,0,0,0,0,0).
type(4,0,0,0,1,0,0,0,0,0,0,0,0,0,0).
type(5,0,0,0,0,1,0,0,0,0,0,0,0,0,0).
...
type(10,0,0,0,0,0,0,0,0,0,1,0,0,0,0).
type(11,0,0,0,0,0,0,0,0,0,0,1,0,0,0).
type(12,0,0,0,0,0,0,0,0,0,0,0,1,0,0).
type(13,0,0,0,0,0,0,0,0,0,0,0,0,1,0).
type(14,0,0,0,0,0,0,0,0,0,0,0,0,0,1).

}

Program 4: Extended logic program for the predicate `contamination`.

{

$\neg$ `contamination (ID, BL, OF, IA, CA)` $\leftarrow$
`not(contamination (ID, BL, OF, IA, CA),`
`not(abducible`$_{contamination}$ `(ID, BL, OF, IA, CA)).`

contamination(1,1,0,0,0).
contamination(2,0,1,0,0).
contamination(3,0,0,1,0).
contamination(4,0,0,0,1).
contamination(5,0,0,0,0).
...
contamination(12,1,1,1,0).
contamination(13,1,0,1,1).
contamination(14,0,1,1,1).
contamination(15,1,1,0,1).
contamination(16,1,1,1,1).

}

2.1.2 Classification

clauses responsible for the waste classification.
every clause has the following representation:

$$\begin{gathered} Condition \\ \downarrow \\ Action \end{gathered}$$

As mencioned above, somewhere, there are four possible classes that will be now represented.

Class 1:

$$\begin{gathered} \text{[waste (ID, Class, SR, TY, CT), SR} < 3\text{, TY} < 3\text{, CT} = 5\text{]} \\ \downarrow \\ \text{[retract(waste(ID, Class, SR, TY, CT)), assert(waste(ID, C1, SR, TY, CT))].} \end{gathered}$$

Class 2:

$$\begin{gathered} \text{[waste (ID, Class, SR, TY, CT), SR} = 6\text{, } 2 < \text{TY} < 7\text{, CT} = 5\text{]} \\ \downarrow \\ \text{[retract(waste(ID, Class, SR, TY, CT)), assert(waste(ID, C2, SR, TY, CT))].} \end{gathered}$$

Class 3:

$$\begin{gathered} \text{[waste (ID, Class, SR, TY, CT), SR} > 2\text{, } 3 < \text{TY} < 10\text{, CT} < 4\text{]} \\ \downarrow \\ \text{[retract(waste(ID, Class, SR, TY, CT)), assert(waste(ID, C3, SR, TY, CT))].} \end{gathered}$$

Class 4:

$$\begin{gathered} \text{[waste (ID, Class, SR, TY, CT), SR} > 4\text{, TY} > 9\text{, CT} = 4\text{]} \\ \downarrow \\ \text{[retract(waste(ID, Class, SR, TY, CT)), assert(waste(ID, C4, SR, TY, CT))].} \end{gathered}$$

Stopping Condition:

$$\text{[waste (ID, Class, SR, TY, CT)]} \rightarrow \text{[print(Class),stop].}$$

3 References

[1] Md. Sohrab Hossain, Nik Norulaini Nik Ab Rahman, Venugopal Balakrishnan, Vignesh R. Puvanesuaran, Md. Zaidul Islam Sarker, and Mohd Omar Ab Kadir. Infectious Risk Assessment of Unsafe Handling Practices and Management of Clinical Solid Waste. *International Journal of Environmental Research and Public Health*, 10(2):556–567, 2013.

[2] Mahmut AKBOLAT, Cemile DEDE, Oguz ISIK, and Hakan SAGLAM. Medical Waste Management Practices in Turkey: A Case Study in Sakarya. *Pakistan Journal of Medical Sciences*, 27(4), 2011.

[3] Issam A Al-Khatib and Chikashi Sato. Solid Health Care Waste Management Status at Health Care Centers in the West Bank–Palestinian Territory. *Waste Management*, 29(8):2398–403, 2009.

[4] Akinwale Coker, Abimbola Sangodoyin, Mynepalli Sridhar, Colin Booth, Paul Olomolaiye, and Felix Hammond. Medical Waste Management in Ibadan, Nigeria: Obstacles and Prospects. *Waste Manag*, 29(2):804–11, 2009.

[5] Magda M. Abd El-Salam. Hospital Waste Management in El-Beheira Governorate, Egypt. *Journal of Environmental Management*, 91(3):618–629, 2010.

[6] Felicia Nemathaga, Sally Maringa, and Luke Chimuka. Hospital Solid Waste Management Practices in Limpopo Province, South Africa: a Case Study of Two Hospitals. *Waste Management*, 28(7):1236–1245, 2008.

[7] José Neves, Jorge Ribeiro, Paulo A. A. Pereira, Victor Alves, José Machado, António Abelha, Paulo Novais, César Analide, Manuel Santos, and Manuel Fernández-Delgado. Evolutionary Intelligence in Asphalt Pavement Modeling and Quality-of-Information. *Progress in Artificial Intelligence*, 1(1):119–135, 2012.

[8] Anabela Botelho. Achieving Compliance With Healthcare Waste Management Regulations: Empirical Evidence From Small European Healthcare Units. 2011.

[9] Gestão de Resíduos Hospitalares para Unidades de Cuidados Continuados Integrados. Technical report, Ministério da Saúde, 2011.

[10] José Alferes and Luís Pereira. *Reasoning With Logic Programming*. Lecture Notes in Artificial Intelligence. Springer, 1111 edition, 1996.

[11] José Neves. A Logic Interpreter to Handle Time and Negation in Logic Data Bases, booktitle = Proceedings of the 1984 Annual Conference of the ACM on The Fifth Generation Challenge, series = ACM '84, year = 1984, isbn = 0-89791-144-X, pages = 50–54, numpages = 5, url = http://doi.acm.org/10.1145/800171.809603, doi = 10.1145/800171.809603, acmid = 809603, publisher = ACM, address = New York, NY, USA, keywords = Assumption, Logic data base, Negative information, Open world, Prolog, Query-by-example, Time stamping,.

[12] Marc Denecker and Antonis C. Kakas. Abduction in Logic Programming. In *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part I*, pages 402–436, London, UK, UK, 2002. Springer-Verlag.

[13] José Neves, José Machado, César Analide, António Abelha, and Luís Brito. The Halt Condition in Genetic Programming. In José Neves, Manuel Filipe Santos, and José Machado, editors, *Progress in Artificial Intelligence, 13th Portuguese Conference on Aritficial Intelligence, EPIA 2007, Workshops: GAIW, AIASTS, ALEA, AMITA, BAOSW, BI, CMBSB, IROBOT, MASTA, STCS, and TEMA, Guimarães, Portugal, December 3-7, 2007, Proceedings*, volume 4874 of *Lecture Notes in Computer Science*, pages 160–169. Springer, 2007.

ID_Waste	Class	ID_Source	ID_Type	ID_Contamination
1	C	6	1	1
2	C	4	2	5
3	C	6	4	5
4	C	3	6	{2, 3}
5	C	{5, 6}	3	4
6	C	$\perp$	1	5

ID_Source	GS	SS	WS	HS	LS	MS
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	0	0	1	0	0	0
4	0	0	0	1	0	0
5	0	0	0	0	1	0
6	0	0	0	0	0	1

ID_Type	GP	FW	CP	OM	DP	IP	MW	NB	SW	IB	MD	CH	AC	CO
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0
			...				...				...			
10	0	0	0	0	0	0	0	0	0	1	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1

ID_Contamination	BL	OF	IA	CA
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	0	0	0	0
	...		...	
12	1	1	1	0
13	1	0	1	1
14	0	1	1	1
15	1	1	0	1
16	1	1	1	0

Figure 3: Database model.