Supply Chain Tracing of Multiple Products under Uncertainty and Incomplete Information
An Application of Answer Set Programming

Monica L. Nogueira and Noel P. Greis
Kenan-Flagler Business School, The University of North Carolina at Chapel Hill, Kenan Center CB#3440, Chapel Hill, NC, 27599 U.S.A.

Keywords: Answer Set Programming, Knowledge Representation, Supply Chain, Traceability, Food Recall Process.

Abstract: Food supply chains are complex networks involving many organizations and food products from the farm to the consumer. The ability to quickly trace the trajectory of a tainted food product and to identify the origin of the contamination is essential to minimizing the economic and human costs of foodborne disease. Complexities arise when multiple products traverse multiple states and/or countries and when products cross multiple intersecting supply chains. In this paper we use the example of a recent Salmonella contamination involving tomatoes and peppers imported from Mexico into the U.S. to demonstrate the use of Answer Set Programming to localize the source of contamination in a complex supply chain characterized by uncertainty and incomplete information.

1 INTRODUCTION

Traceability of food products through the supply chain is a key problem for public health officials worldwide as supply chains have become larger, more complex, and increasingly span both national and international jurisdictions. Globalization, international trade, and new eating habits, are contributing factors to a reduced shelf life and a potentially higher risk of contamination. Control of scale and scope of a foodborne illness outbreak is directly linked to authorities' ability to track and trace tainted products in the supply chain quickly and accurately thus ensuring their removal and disposal before further harm occurs. But even in developed countries no single traceability scheme has been widely adopted and adequate track and trace methods are yet to be identified (Fritz and Schiefer, 2008; Regattieri et al., 2007).

This paper addresses the challenge of tracing multiple food products across complex supply chains by extending previous work of (Nogueira and Greis, 2013) that demonstrated the use of the logic programming approach Answer Set Programming (ASP) (Marek and Truszczynski, 1999) to the food safety domain. Our contribution herein is to apply ASP to solve other common practical traceability problems motivated by a past multiple product outbreak event in the United States—the 2008 tomatoes and peppers outbreak (CDC, 2008). First, we demonstrate how to compute the effects of a food recall and its impact on firms and states, and how to trace products within a geographically targeted supply chain. Second, we trace concurrently recalled (multiple) products through diverse food chains and identify any points of intersection between these chains. Third, we simulate, or "probe," the tracing of suspected tainted products before confirmation or any recalls are issued, and identify firms that may be potentially involved in the outbreak.

Section 2 discusses motivation for this research and related work. Section 3 defines a complex supply chain and its encoding in ASP. Section 4 shows how to compute effects of a recall within a geographic area through the use of aggregate functions of the ASP solver DLV. Section 5 presents our ASP-based solution to the traceability problems linking multiple products and supply chains. Section 6 concludes the paper and presents future research.

2 MOTIVATION AND RELATED WORK

The U.S. Centers for Disease Control and Prevention

L. Nogueira M. and P. Greis N..
Supply Chain Tracing of Multiple Products under Uncertainty and Incomplete Information - An Application of Answer Set Programming.
DOI: 10.5220/0004627603990406
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2013), pages 399-406
ISBN: 978-989-8565-81-5
Copyright © 2013 SCITEPRESS (Science and Technology Publications, Lda.)

(CDC) define a foodborne disease outbreak as two or more illnesses caused by consumption of a common food. Annually 48 million Americans, fall ill due to a foodborne illness (Scallan et al., 2011). The number of outbreaks reported to CDC by state health officials for 2009 and 2010 (1,527 in total) show that no food commodity was identified for approximately 58% (or 892) outbreaks (CDC, 2013). Due to delays of one to two weeks between the consumption of a tainted food and the onset of illness, patients usually do not remember the foods they ate. Hence, new methods must be developed that can deal with incomplete information yet still generate candidate sources of contamination and thereby reduce the burden of illness and the economic impact to populations and businesses.

Traceability refers broadly to the ability, for any product at any stage within the food chain, to identify the initial source (backward tracing) and, eventually, its final destination (forward tracing) (Fritz and Schiefer, 2009). Tracking refers to the ability to identify, for any product, its actual location at any given time. Together these two capabilities provide the functionality of a “track-and-trace” system for the food supply chain. In this work we tackle the traceability problem by utilizing a logic formalism to encode and generate likely trajectories for contamination due to a recall and to identify all possibly affected companies and their products.

In this paper we model the recent “Tomatoes and Jalapeño Peppers” outbreak. From May to August 2008, the CDC recorded a large foodborne disease outbreak with 1,442 cases, counting 286 hospitalizations and possibly two deaths, affecting 43 states, the District of Columbia, and Canada. Several case-control studies conducted to identify the contamination source(s) produced mixed results, pointing to jalapeño peppers as a major transmission vehicle, and serrano peppers and tomatoes as other possible vehicles (CDC, 2008). In July 2008 jalapeño peppers were traced by FDA to a farm in Mexico which also grew serrano peppers. For encoding simplicity, different types of tomatoes or peppers are not considered here. Lack of publicly available data on international food firms, led us to restrict our representation to the U.S. supply chain.

In the last decade, developed countries have approved more stringent legislation to improve the safety standards of the food they produce (Kher et al., 2010; McEntire and Bhatt, 2012). However, the quality and safety of food products produced in developing countries are less regulated and, thus, in many cases more lax. Currently supply chains cross developed and developing countries which complicates traceability requirements. Hence, traceability methods based on unique identification schemes must work globally, and interoperability of radio frequency identification-based and other barcode schemes must be seamless (GS1, 2010; Thakur et al., 2011). Our approach to the traceability problem is independent of the availability of such identifiers. We employ publicly available information about food businesses, e.g. type of products sold and company role in the supply chain, in conjunction with food ontologies, to trace possible trajectories paths of food distribution and narrow down affected firms and areas.

3 REPRESENTING COMPLEX SUPPLY CHAINS IN ASP

A supply chain is the entire network connecting, directly or indirectly, different companies that participate in the coordinated production, manufacture, handling, distribution and/or retail of a specific product to fulfill a customer request. Specifically, a food supply chain consists of all the steps required to transform a (raw) food commodity into a product ready for consumption. An illustrative example of a generic, complex supply chain crossing national borders is shown in Figure 1. The food chain encompasses farmers or growers of raw food commodities, processors or manufacturers who convert raw food into products, wholesalers that store and commercialize products, and distributors that deliver it to retailers who sell the product directly to consumers, as well as brokers, importers, and exporters. A supply chain is dynamic and there is a constant flow of product, information, and capital between its stakeholders. This directed flow is represented in Figure 1 by the arrows linking the different company types. This abstract view of a supply chain serves as the basis for building our ASP representation and program to solve common, practical traceability problems.

3.1 ASP Basic Syntax and Semantics

The ASP paradigm is based on the stable models/ answer sets semantics of logic programs (Gelfond and Lifschitz, 1988, 1991) and has been shown to be a powerful formalism for knowledge representation including defaults, inheritance reasoning, reasoning about actions and their effects, and to be particularly useful in solving challenging search problems. ASP reduces search problems to the computation of the

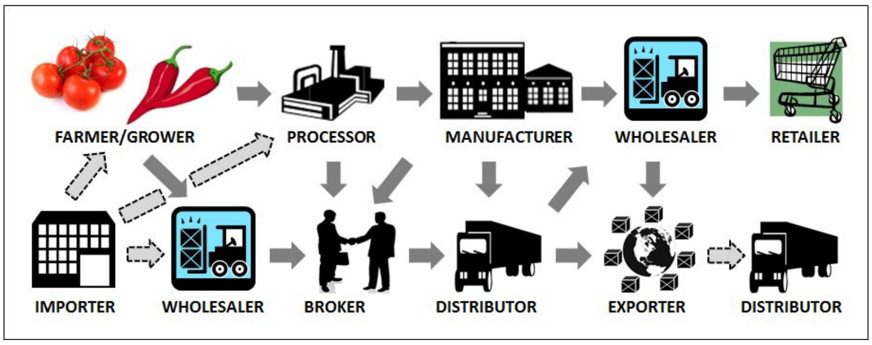

Figure 1: An Example of a Generic, Complex Supply Chain Crossing National Borders.

stable models of the problem and a growing number of ASP solvers — programs that generate the stable models of a given problem encoded in the ASP formalism, e.g. clasp, DLV, Pbmodels, Smodels, etc., are available. Syntax and semantics of the language is found in (Gelfond and Lifschitz, 1991).

Language signature Σ contains predicates, constants, and function symbols. Terms and atoms are formed as habitual in first-order logic. A literal is either an atom (positive literal) or an atom preceded by ¬ (classical or strong negation), a negative literal. Literals l and $\neg l$ are called contrary. Ground literals and terms are those not containing variables. A consistent set of literals does not contain contrary literals. The set of all ground literals is denoted by *lit*(Σ). A rule is a statement of the form:

$$h_1 \vee ... \vee h_k \leftarrow l_1, ..., l_m, \text{not } l_{m+1}, ..., \text{not } l_n. \qquad (1)$$

where h_i's and l_i's are ground literals, *not* is a logical connective called negation as failure or default negation, and symbol ∨ is the disjunction operator. The rule head appears to the left of symbol ←, and the body on its right side. Intuitively, the rule means that if a reasoner believes $\{l_1, \ldots, l_m\}$ and has no reason to believe $\{l_{m+1}, \ldots, l_n\}$, then it must believe one of the h_i's. If the head is replaced by ⊥ (falsity) then the rule is called a constraint. The intuitive meaning of a constraint is that its body must not be satisfied. Rules with variables (denoted by capital letters) are used as a short hand for the sets of their ground instantiations. An ASP program is a pair of ⟨Σ, Π⟩, where Σ is a signature (usually implicit) and Π is a set of rules (a program) over Σ. A stable model (or answer set) of a program Π is one of the possible sets of literals of its computable consequences under the stable model/answer set semantics.

Our encoding, the set of rules of program Π, contains roughly 50 rules, while there are thousands of records—in ASP, rules with an empty body, also called "facts"—corresponding to the database of companies, and food and geographic ontologies. The ASP solver used is DLV (Calimeri et al., 2002). The solver is tasked with computing the set of firms in a target area affected by a single or multiple recalls, and firms that could be affected when more than one product is suspected but no recall has been issued. Advantages of encoding the food supply chain traceability problem as an ASP program include: (1) ASP allows easy encoding of many forms of domain knowledge and generating hierarchical ontologies for diferent types of domain relevant information, e.g. geographical, disease (Nogueira and Greis, 2011). Encoding of heuristics makes it possible to prune the search space and increase the efficiency of tracking and tracing a contaminated product in the supply chain; (2) Modular ASP programs, where each module will or will not be executed depending on the solution sought, provide added flexibility for execution that helps control innefficiency and avoid combinatorial explosion; (3) ASP is well-suited to represent action and change. A food supply chain is an intrinsically dynamic environment where food products move from one node to the next in the chain, and the track-and-trace of contaminated products posing risk to human lives should be highly efficient to curb a contamination event that may spread very rapidly; and (4) ASP is well-suited to deal with incomplete information—an inherent problem of this domain as food enterprises are averse to sharing information about their supplier and customer bases which represents competitive advantage to their business.

3.2 Supply Chain as an ASP Program

We leverage publicly available information on U.S. food companies to build the database of entities that form our supply chain. Since food companies may have multiple roles in the chain and produce more than one product, they are represented in our ASP program by rules of form (2)-(4). A food recall by a given company is encoded as (5). A conceptual model for supply chains is that of an acyclic, directed graph as illustrated in Figure 1. Knowledge about supply chain operations provides the semantics for the graph's vertices and edges. Each vertex, except for a point of origin or food source such as the grower or importer, corresponds to a firm A that performs a transformative step on a less processed product to generate a more refined product supplied to firm B located down-stream in the chain and to whom A is directly connected by an edge in the graph. Clearly, if A supplies B—who in turn supplies another firm or a final consumer, then their products are derived from the same food commodity. Rule (6) encodes the supply functions.

Each edge of the graph represents a valid flow of ingredients or final product(s) for resale from a firm A set upstream in the chain and directly connected to a firm B by an edge in the graph, i.e. a supply operation between the two types of entities connected in that supply chain. Figure 1 contains 17 edges with 4 dashed edges corresponding to foreign supplying functions. The U.S. supply chain defined by 13 solid edges is encoded by 13 individual rules of the form of (7) or (8), depending on whether an ingredient or final product is supplied. We omit some of the rules to avoid repetition and save space. Tracing products forward, and similarly backward, in the supply chain is achieved by (9)-(14), which also compute the transitive closure of these relations. Our knowledge base contains a simple ontology which models the main stages of a food product as it evolves from raw, unprocessed food at the farmer/grower level of the supply chain to a processed food ready for consumption at the retail point-of-sale. The ontology is built with rules of form (15)-(16) for each food supply chain (47) to represent a product's primary ingredient(s) and its derivative products as illustrated by facts (17)-(46). The ASP program described and below is based on work from (Nogueira and Greis, 2012 and 2013).

```
firm(Idcode,Name,State).                  (2)
type_firm(Idcode,Type).                   (3)
prod_supplied(Idcode,Product).            (4)
recall(Product,Idcode).                   (5)

supplies(A,I1,B) :-                       (6)
 valid_supply(F,A,B),
 is_of(I1,F),prod_supplied(A,I1),
 is_of(I2,F),prod_supplied(B,I2).
valid_supply(F,A,B) :-                    (7)
 type_firm(A,grower),
 type_firm(B,processor),A!=B,
 sc(F),is_ingr(I1,I2),
 is_of(I1,F),prod_supplied(A,I1),
 is_of(I2,F),prod_supplied(B,I2).

valid_supply(F,A,B) :-                    (8)
 type_firm(A,grower),
 type_firm(B,wholesaler),A!=B,
 sc(F),prod_supplied(A,I),
 is_of(I,F),prod_supplied(B,I).

fwtrace(C,LC,F,A,LA) :-                   (9)
 recall(F,C),supplies(C,F,A),
 firm(C,_,LC),firm(A,_,LA),  C!=A.

fwtrace(B,LB,F1,A,LA) :-                  (10)
 is_ingr(F,F1),supplies(B,F1,A),
 fwtrace(C,LC,F,B,LB),
 firm(C,_,LC),firm(B,_,LB),
 firm(A,_,LA),B!=C,B!=A,A!=C.

fwtrace(B,LB,F,A,LA) :-                   (11)
 supplies(B,F,A),
 fwtrace(C,LC,F,B,LB),
 firm(B,_,LB),firm(A,_,LA),
 firm(C,_,LC),B!=C,B!=A,A!=C.

bktrace(A,LA,F,C,LC) :-                   (12)
 recall(F,C),supplies(A,F,C),
 firm(C,_,LC),firm(A,_,LA),C!=A.

bktrace(B,LB,F1,C,LC):-                   (13)
 is_ingr(F1,F),supplies(B,F1,C),
 bktrace(C,LC,F,A,LA),
 firm(B,_,LB),firm(C,_,LC),
 firm(A,_,LA),B!=C,B!=A,A!=C.

bktrace(B,LB,F,C,LC) :-                   (14)
 supplies(B,F,C),
 bktrace(C,LC,F,A,LA),
 firm(B,_,LB),firm(C,_,LC),
 firm(A,_,LA),B!=C,B!=A,A!=C.

is_of(Product,Chain).                     (15)
is_ingr(Product1,Product2).               (16)

is_of(ttfresh, tomatoes).                 (17)
is_of(ttketchup, tomatoes).               (18)
is_of(ttpastepuree, tomatoes).            (19)
…
is_of(pepper, peppers).                   (27)
is_of(ppfresh, peppers).                  (28)
is_of(ppchili, peppers).                  (29)

is_ingr(ttfresh, ttpastepuree).           (30)
is_ingr(ttpreserved, ttketchup).          (31)
is_ingr(ttpreserved, ttsauce).            (32)
                                          …
is_ingr(ppfresh, pepper).                 (43)
is_ingr(ppgreenfrsh, ppcrshgrd).          (44)
is_ingr(ppcrshgrd,ppdrycrshgrd).          (45)
is_ingr(ppchili, ppdrycrshgrd).           (46)

supply_chain(Chain).                      (47)
supply_chain(tomatoes).           (48)
supply_chain(peppers).            (49)
```

4 AGGREGATE FUNCTIONS TO COMPUTE RECALL EFFECTS

Assume that a firm, identified by code "cp58" in our firms' database, has a dual role of produce grower and processor and recalls its fresh tomatoes product as denoted by fact (50) added to the program. The solution for this traceability problem contains 4,132 atoms "fwtrace" and "bktrace," with 3,255 possible paths of contamination forward in the chain, and 877 backward from the point of recall. A portion of the solution for this tracing exercise appears below.

```
 recall(ttfresh,cp58).                (50)
{fwtrace(cp58,ca,ttfresh,cp14,az),
fwtrace(cp58,ca,ttfresh,cp112,ca),…
bktrace(cp58,ca,ttfresh,cp431,ca),
bktrace(cp58,ca,ttfresh,cp755,az),…}
```

4.1 Track and Trace of Single Recalled Products

An important question for public health officials is to determine how many firms are affected by this recall forward in the supply chain, i.e. how many received the recalled product. DVL authors have extended its language to provide constructs that enable arithmetic operations over a set of atoms, like sum and count, and allow answering such questions. Incidentally, other ASP solvers, e.g. Smodels, provide comparable constructs with slightly different semantics and syntax.

Hence, the language described in Section 3.1 is extended by sets, aggregate functions, atoms, and literals as defined in (Dell'Armi et al., 2003). A set is a pair of the form {*T:Conj*}. In a symbolic set, *T* is a list of variables and *Conj* is a conjunction of standard literals. In a ground set, *T* consists of a list of constants, and *Conj* is ground (variable free). Aggregate functions are of the form *f(S)*, where *S* is a set and *f* is a *function name* among #count, #min, #max, #sum, #times. An aggregate atom is formed by aggregate functions, written as $L_g \prec_1 f(S) \prec_2 R_g$, where $\prec_1, \prec_2 \in \{=, <, \leq, >, \geq\}$, and L_g and R_g are terms (one being possibly omitted). Atoms can be either standard or aggregate, and literals constructed from aggregate atoms are aggregate literals.

Rule (51) shows the use of the aggregate function "#count{T:Conj}=N" to answer the pending question and to compute how many N firms are affected forward in the supply chain by the recall of fresh tomatoes from firm "cp58". The solution computed by the DLV solver shows that overall 94 firms have received the contaminated product directly from company "cp58", or indirectly from other firms located more than one step forward in the chain. Rule (52) computes how many firms received the contaminated product directly from recalling firm "cp58". The DLV solver finds that 56 of the 94 firms were directly supplied by "cp58". A natural question is how many states are affected by this recall, i.e. in how many states are these 94 firms located. Rule (53) performs this computation resulting in 23 states. Similarly, rule (54) computes the number of states where the 56 firms directly supplied by company "cp58" are located (12 states). The solution computed by the DLV solver for this traceability problem includes 877 atoms of type "bktrace". Similarly to (51)-(54), (55)-(58) count the number of firms and states in the contamination path backward in the supply chain to firm "cp58".

```
all_ftrace_firms(N) :-               (51)
 #count{C:fwtrace(_,_,_,C,_)}=N.

dir_ftrace_firms(N) :-               (52)
 recall(_,C),
 #count{F:fwtrace(C,_,_,F,_)}=N.

all_ftrace_states(N) :-              (53)
 #count{S:fwtrace(_,_,_,_,S)}=N.

dir_ftrace_states(N) :-              (54)
 recall(_,C),
 #count{S:fwtrace(C,_,_,_,S)}=N.

all_btrace_firms(N) :-               (55)
 #count{F:bktrace(F,_,_,_,_)}=N.

dir_btrace_firms(N) :-               (56)
 recall(_,C),
 #count{F:bktrace(F,_,_,C,_)}=N.

all_btrace_states(N) :-              (57)
 #count{S: bktrace(_,S,_,_,_)}=N.

dir_btrace_states(N) :-              (58)
 recall(_,C),
 #count{S:bktrace(_,S,_,C,_)}=N.
```

In supply chains with multiple products, firms may have more than one role. Thus, the intersection of firms computed by (51)-(52) and (55)-(56) may be greater than zero. For this reason, we must eliminate duplicates as computed by rules (59)-(62). Similar rules compute the number of firms directly linked to "cp58" upstream and downstream in the supply chain. In fact, 98 unique firms in 25 states are affected by this recall since 32 firms in 5 states belong to the intersection.

```
total_aff_firms(T) :-                    (59)
 #count{A:fwtrace(_,_,_,A,_)}=M,
 #count{C:bktrace(C,_,_,_,_)}=N,
 firms_intersect(I),U=M+N,T=U-I.

firms_intersect(I) :-                    (60)
 #count{A:fwtrace(_,_,_,A,_),
      bktrace(A,_,_,_,_)}=I.

total_aff_states(T) :-                   (61)
 #count{LA:fwtrace(_,_,_,_,LA)}=M,
 #count{LC:bktrace(_,LC,_,_,_)}=N,
 states_intersect(I),U=M+N,T=U-I.

states_intersect(I) :-                   (62)
 #count{L:fwtrace(_,_,_,_,L),
      bktrace(_,L,_,_,_)}=I.
```

4.2 Geographically Targeted Tracing

Regional and state public health officials are tasked with the coordination of recall efforts occurring in the geographical area within their jurisdictions. Hence, it is important to obtain specific information about affected companies located within the boundaries of such areas. For this reason, our ASP program encodes four U.S. regions, atoms (63)-(66), and 50 states plus the District of Columbia, with 51 (ground) atoms of type (67). A more complete geographic ontology was developed in (Nogueira and Greis, 2011). Regions of interest are indicated to the program by adding (ground) atoms of type (68). Rules (69)-(70) make it possible to generate a list of all potentially affected firms in a given region of the country to assist the efforts of its regional recall coordinators. Identifying firms downstream from the point of recall, i.e. forward tracing, is performed by (69), and upstream, or backward, tracing by (70).

The number of states affected by a recall within the trace region (68) is computed by (71), and those belonging to the intersection by (72). The recall of fresh tomatoes issued by "cp58" can potentially affect 7 (out of 17) states in the South, 5 (out of 9) states in the Northeast, 6 (out of 12) in the Midwest, and 7 (out of 13) in the West; or 25 states total. Similarly, (73)-(74) compute the number of firms affected by the recall within the specified region.

From all 98 unique firms affected, the program returns 16 firms located in the Northeast, 21 in the South, 25 in the Midwest, and 36 in the West. A complete list of supplier and customer firms sharing a contamination path, and located within the same region, is generated by (75)-(76). For this recall, 21 such firms are in the South, 36 in the West, 15 in the Northeast, and 22 in the Midwest, i.e. 94 in total.

```
us_region(northeast).                    (63)
us_region(midwest).                      (64)
us_region(south).                        (65)
us_region(west).                         (66)
us_state(State,Region).                  (67)

geotrace(Region).                        (68)

ftrace_firms(R,A,LA) :-                  (69)
 geotrace(R),us_state(LA,R),
 fwtrace(C,LC,F,A,LA).

btrace_firms(R,C,LC) :-                  (70)
 geotrace(R),us_state(LC,R),
 bktrace(C,LC,F,A,LA).

reg_states_aff(R,T) :-                   (71)
 #count{LA:ftrace_firms(R,_,LA)}=M,
 #count{LB:btrace_firms(R,_,LB)}=N,
 reg_states_intersect(R,I),
 U=M+N,T=U-I.

reg_states_intersect(R,N) :-             (72)
 geotrace(R),
 #count{L:ftrace_firms(R,_,L),
      btrace_firms(R,_,L)}=N.

reg_firms_aff(R,T) :-                    (73)
 #count{A:ftrace_firms(R,A,_)}=M,
 #count{B:btrace_firms(R,B,_)}=N,
 reg_firms_intersect(R,I),
 U=M+N,T=U-I.

reg_firms_intersect(R,N) :-              (74)
 geotrace(R),
 #count{A:ftrace_firms(R,A,_),
      btrace_firms(R,A,_)}=N.

in_region(R,A,LA) :-                     (75)
 geotrace(R),fwtrace(C,LC,F,A,LA),
 us_state(LA,R),us_state(LC,R).

in_region(R,C,LC) :-                     (76)
 geotrace(R),bktrace(C,LC,F,A,LA),
 us_state(LA,R),us_state(LC,R).
```

5 TRACK-TRACE OF MULTIPLE CONTAMINATED PRODUCTS

There are always multiple on-going cases of food contamination and associated food recalls being investigated in a country or region. Hence, any proposed solutions must be able to track and trace multiple tainted products in the supply chain. Public health officials often seek information regarding several outbreaks at the same time. We examine two common situations involving multiple recalls next.

5.1 Tracing Concurrently Recalled Products

To demonstrate that this ASP program can be used to trace multiple products, let us assume that recalls have been issued for the two main products involved in the 2008 tomatoes and jalapeño peppers outbreak, and that the supply chains of these products closely resemble that illustrated by Figure 1. Without loss of generality we assume that the peppers supply chain follows the linear structure captured by rules of type (7) or (8) corresponding to the top portion of Figure 1. To complete this description we need only to ensure that the food ontology employed by the ASP program contains: a) facts (17)-(46) describing the production hierarchy for tomato and pepper products, and b) recall (77) of peppers issued by the grower/processor/exporter firm "cp951". ASP programs that require only small changes to accommodate new circumstances, as this one does, are called "elaboration tolerant". Programs exhibiting such property are highly preferred because they require less time and effort to be modified to solve larger, more complex problems.

The new, augmented ASP program will now take into account both supply chains when computing the firms affected by these recalls. Public health officials may be especially interested in firms that produce both products being recalled. A quick, albeit naive, way to list these firms can be achieved with (78). The number of firms that produce both products, as well as the number of states where these firms are located, can be calculated by (79). DLV results found 72 companies in 19 states potentially affected by the contamination of both products.

```
 recall(ppfresh,cp941).                (77)

 dir_aff_firms(C,L) :-                 (78)
    firm(C,_,_,L),
recall(P1,_),prod_supplied(C,P1),
recall(P2,_),prod_supplied(C,P2).

 recall_eff_totals(C,S) :-             (79)
    #count{A:dir_aff_firms(A,_)}=C,
    #count{L:dir_aff_firms(_,L)}=S.
```

5.2 Tracing Products Suspected of Contamination before Confirmation

An equally important but different situation faced by health officials concerns tracing several products suspected of contamination before clinical test results or epistemological studies provide definite confirmation of the tainted product. When there is a lack of evidence due to uncertainty and incomplete information, public safety must be ensured and provisory warnings must alert the population to avoid consumption of suspected products. In these cases, a software tool that allows scenario-based reasoning can help identify possible contamination paths through all the supply chains involved. Thus, we introduce disjunctive rules to our program as an effective construct to simulate what-if scenarios—each corresponding to a stable model of the ASP program—and show how to interpret such models.

Assume that during an on-going contamination outbreak, a number of patients reported consuming both fresh tomatoes and peppers weeks before the onset of their illness. Hence, officials may suspect that one of these foods is the cause of the outbreak and, until tests can prove which one, those firms producing both products must be investigated. The traceability problem is thus reduced to identifying the firms that produce the suspected products. A rule of type (80), e.g. (81), expresses the suspicion that one of two products is tainted and specifies what type of firms may be part of the contamination. Since no recall has been issued, we retract any rules of the form (5) and, to keep using the existing ASP program, we redefine predicates "recall" and "supply_chain" in terms of new predicates "recalled(P,C)" and "suspect(P,T)" with (82)-(85). Consequently, whenever a food recall is issued facts of type "recalled(P,C)" are added to the program.

```
 suspect(A,T1) v suspect(B,T2).     (80)

  suspect(ttfresh,processor) v      (81)
suspect(ppfresh,grower).

  recall(P,C) :- recalled(P,C).     (82)

  recall(P,C) :-                    (83)
     firm(C,_,_,L),type_firm(C,T),
     suspect(P,T),prod_supplied(C,P),
     us_state(L,R),geotrace(R).

  supply_chain(S) :-                (84)
     recalled(P,_),is_of(P,S).

  supply_chain(S) :-                (85)
     suspect(P,_),is_of(P,S).
```

The DLV solver derives two solutions, i.e. stable models, for this program corresponding to the two possible contamination scenarios. The meaning of atoms "recall(P,C)" in the solutions below is that firm C, located in the tracing geographic area(s), produces suspected product P. For space reasons, we omit the complete solution of affected firms traced in the supply chain encompassing all four U.S. regions. Lastly, the forward and backward traces are done in two separate program runs to avoid the combinatorial explosion that occurs when searching for all affected firms without starting from a single point of recall.

```
{suspect(ttfresh,processor),
 supply_chain(tomatoes),
 recall_eff_totals(39,8),
 recall(ttfresh,cp174),
 recall(ttfresh,cp431),…}

{suspect(ppfresh,grower),
 supply_chain(peppers),
 recall_eff_totals(48,17),
 recall(ppfresh,cp239),
 recall(ppfresh,cp753),…}
```

6 CONCLUSIONS

This paper demonstrates the utility of answer set programming in identifying the contamination source(s) in complex food chains characterized by multiple products flowing through intersecting supply chains. Using rules and aggregate functions, we compute the effects of recalls on targeted geographic areas to identify the affected companies located within. Disjunctive rules are used to simulate possible scenarios in the face of uncertainty and incomplete information. We use the example of the 2008 food contamination event involving tomatoes and jalapeños to show the value of this approach to state agencies charged with managing product recalls in the event of a foodborne disease outbreak.

Two future avenues of research include: 1) moving to a risk-based approach by adding knowledge extracted from publicly available data on reported foodborne illness such as the likelihood that a particular food has a high potential risk for contamination, or the degree of severity of illness attributed to a particular food; and 2) incorporating supply chain data on international food businesses.

REFERENCES

Calimeri, F; Dell'Armi, T; Eiter, T; Faber, W; Gottlob, G; Ianni, G; Ielpa, G; Koch, C; Leone, N; Perri, S; Pfeifer, G; Polleres, A; 2002. "The dlv system." In Flesca, S; Ianni, G; eds., *JELIA'02,* pp. 537–540.

CDC, 2013. "Surveillance for Foodborne Disease Outbreaks—U.S., 2009–10," *MMWR 2013*, 62:3, pp.41-47.

CDC, 2008. "Outbreak of *Salmonella* Serotype Saint-paul Infections Associated with Multiple Raw Produce Items—U.S.," *MMWR 2008*, 57:34, pp. 929-934.

Dell'Armi, T; Faber, W; Ielpa, G; Leone, N; Pfeifer, G; 2003. "Aggregate Functions in DLV." In Answer Set Programming: Adv. in Theory and Implementation, vol. 78, CEUR Workshop Proceedings, pp. 274–288.

Fritz, M; Schiefer, G; 2008. "Tracking and tracing in food networks." In *IAALD AFITA WCCA'08*, Tokyo University of Agriculture, pp. 967–972.

Fritz, M; Schiefer, G; 2009. "Tracking, tracing and business process interests in food commodities: A multi-level decision complexity." *International Journal Production Economics*, 117:2, pp. 317–329.

Gelfond, M; Lifschitz, V; 1988. "The stable model semantics for logic programming." In Kowalski, R; Bowen, K; eds., *International Logic Programming Conf. and Symp.*, pp. 1070–1080. MIT Press.

Gelfond, M; Lifschitz, V; 1991. "Classical negation in logic programs and disjunctive databases." *New Generation Computing*, 9, pp. 365–385.

GS1, 2010. "Business Process and System Requirements for Full Supply Chain Traceability." GS1 Global Traceability Standard, Issue 1.2.2.

Kher, SV; Frewer, LJ; Jonge, JD; Wentholt, M; Davies, OH; Luijckx, NBL; Cnossen, HJ; 2010. "Experts' perspectives on the implementation of traceability in Europe," *British Food Journal*, 112:3, pp. 261-274. Emerald Group Publishing Limited.

Marek, VW; Truszczynski, M; 1999. The Logic Programming Paradigm: a 25-Year Perspective, chap. "Stable models and an alternative logic programming paradigm," pp. 375–398. Springer Verlag, Berlin.

McEntire, J; Bhatt, T; 2012. "Pilot Projects for Improving Product Tracing along the Food Supply System–Final Report." *Institute of Food Technologists*.

Nogueira, ML; Greis, NP; 2011. "Rule-Based Complex Event Processing for Food Safety and Public Health." In Bassiliades, N; Governatori, G; Paschke, A; eds., *RuleML-Europe'11*. LNCS 6826, pp. 376–383. Springer.

Nogueira, ML; Greis, NP; 2012. "Recall-driven Product Tracing and Supply Chain Tracking using Answer Set Programming." In Filipe, J; Dietz, JLG; eds., *KEOD'12*, pp.125–133. SCITEPRESS.

Nogueira, ML; Greis, NP; 2013. "An Answer Set Programming Solution for Supply Chain Traceability." In Filipe, J; Dietz, JLG; eds., *KEOD'12*. Springer.

Regattieri, A; Gamberi, M; Manzini, R; 2007. "Traceability of food products: general framework and experimental evidence." *J Food Eng*, 81, pp. 347–356.

Scallan, E; Hoekstra, RM; Angulo, FJ; Tauxe, RV; Widdowson, M-A; Roy, SL; Jones, JL; Griffin, PM; 2011. "Foodborne Illness Acquired in the United States." *Emerging Infectious Disease*, 17:1, pp. 7–15.

Thakur, M; Sørensen, C-F; Bjørnson, FO; Forås, E; Hurburgh, CR; 2011. "Managing food traceability information using EPCIS framework." *Journal of Food Engineering*, 103:4, pp. 417–433.