An Answer Set Programming Solution for Supply Chain Traceability

Monica L. Nogueira and Noel P. Greis

Center for Logistics and Digital Strategy, The University of North Carolina at Chapel Hill,
Kenan Center CB#3440, Chapel Hill, 27713 U.S.A.
{monica_nogueira,noel_greis}@unc.edu

Abstract. Developing measures to improve the traceability of contaminated food products across the supply chain is one of the key provisions of the 2011 FDA Food Safety Modernization Act (FSMA). In the event of a recall, FSMA requires companies to provide information about their immediate suppliers and customers—what is referred to as "one step forward" and "one step backward" traceability. In this paper we implement the logic-based approach called answer set programming that uses inference rules to trace the flows of contaminated products—both upstream to the source of the contamination and downstream to consumer locations. The approach does not require common standards or unique product identifiers for tracking individual products. This elaboration-tolerant method can accommodate changes in the supply chain such as: 1) the addition of new multiple product pathways; 2) consideration of multiple ingredients in a single product; and 3) multiple products with multiple pathways. We demonstrate this highly flexible methodology for pork and peanut products.

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

1 Introduction

Food safety is a challenging problem that has been growing worldwide due to the globalization of the food supply chain, internationalization of trade, and new eating habits, among other factors. The lack of a consistent, unified, and standardized tracking and tracing system for food manufactured, produced, processed, packed, held, distributed, and sold in the United States is a major pain point of the American food safety system, but this is a problem that affects most countries, if not all. The food supply chain consists of many entities from producer/grower and processor to distributor and retailer. Each of these entities is linked to one another through the food chain. Contamination can enter the food chain at any point due to a range of causes from improper processing or handling to intentional contamination.

In the U.S. once public health officials have determined that a foodborne disease event has occurred and identified the offending product and its manufacturer, a product recall may be issued by the U.S. Food and Drug Administration (FDA)

A. Fred et al. (Eds.): IC3K 2012, CCIS 415, pp. 211–227, 2013.

agency. This recall signals the launch of a series of actions by state food safety departments to remove any contaminated products from retail shelves within their states. State agencies must quickly determine whether any recalled products are being sold by retail enterprises or whether the contaminated products have been used as ingredients in any products being sold. At the point of recall, state agencies are required to piece together information from enterprises across the food chain in an environment where there is not a uniform system for linking this information, nor accepted standards for identifying products, nor any central place where this information is stored and accessible.

The difficulty of the task is complicated by (1) the complexity of the food chain where a single food product can be made of hundreds of ingredients which each may be supplied by multiple suppliers; (2) the fact that uniform standards for data collection in the food industry do not exist, making it difficult to re-create the food chain for contaminated products; and (3) the fact that companies are often reluctant to make public proprietary information about their supply chain suppliers and customers. Further, traceability across enterprise boundaries requires agreements and coordination among suppliers and customers that can be difficult to achieve.

This paper is organized as follows. Section 2 discusses the motivation behind our work. Section 3 presents background information about the problem being solved. Section 4 describes existing traceability schemes. An ASP program encoding the traceability problem for a simple and a more complex supply chain is discussed in Sections 5 and 6, respectively. Conclusions appear in Section 7.

2 Motivation

The lack of track-and-trace capability has received considerable attention recently due to several high-profile and costly incidents of foodborne disease in the United States (c.f. peanut butter, spinach, jalapenos peppers) and abroad (c.f. milk, pork, sprouts). New studies from the U.S. Centers for Disease Control and Prevention (CDC) estimate the total effect of contaminated food consumed in the United Sates as follows: 47.8 million illnesses, 127,839 hospitalizations and 3,037 deaths per year [1, 2]. The total cost of food contamination in the U.S. was recently estimated to be $152 billion a year including health and human welfare costs, as well as economic damage to companies and entire industries [3]. In 2009, the Peanut Corporation of America (PCA) peanut butter contamination alone sickened more than 700 people in 44 states and was associated with nine deaths—and also resulted in the largest dollar-valued food recall in U.S. history. More than 3,000 products were recalled. Early estimates of the costs to the peanut butter industry due to lost peanut butter and peanut sales were more than $1 billion.

The PCA peanut contamination also illustrates the problems of determining both the source and the location of contaminated foods in the food chain. Difficulties are complicated when the contamination is ingredient-driven, that is when the contaminated product is an ingredient in a large number of different products that are sold in many different channels.

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) [4]. 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.

A 2009 traceability exercise conducted by the U.S. Department of Health and Human Services (DHHS) illustrated the gaps in the current system. Investigators purchased 40 different products and attempted to trace each through the supply chain back to the farm or the border, in the event of an imported food. Of the 40 products, only five could be traced back completely to the point of origin; 31 could be traced back only partially; and four of the products could not be traced back at all [5].

3 Background

The FDA Food Safety Modernization Act (FSMA) that was signed into law by President Obama in January 2011 was the first major overhaul of food safety law in the U.S. in decades. It set the stage for a new era in food safety regulation that moves FDA towards new risk based approaches. FSMA includes several key provisions that position the FDA to improve its ability to respond to a food recall. First, the FDA now has the authority to issue a mandatory recall when it has been determined that there is a reasonable probability that a food poses a threat to human health. Previously, FDA could only request a voluntary recall.

FSMA also requires that the FDA establish, as appropriate, within the FDA "a product tracing system to receive information that improves the capacity of the Secretary to effectively and rapidly track and trace food that is in the United States or offered for import into the United States" [6]. FSMA does not specify the details of such a traceability system or the technology to be used, but directs the FDA to conduct at least two pilot projects to evaluate methods for improving traceability. On September 2011, the FDA announced that the Institute of Food Technologists (IFT) will "carry out two new pilot projects at the direction of FDA to explore and demonstrate methods for rapid and effective tracking and tracing of food, including types of data that are useful for tracing, ways to connect the various points in the supply chain and how quickly data can be made available to FDA" [7].

In addition, FSMA expands the registration requirements established by the U.S. Congress in the 2002 Bioterrorism Act that required all facilities that manufacture, process or pack food to register with the FDA, but exempted farms and retail food establishments, by limiting that exemption only to family and smaller growers.

Finally, in support of traceability, FSMA requires companies to provide for all food products "one step forward" and "one step backward" traceability. Food facilities are not required to provide full traceability for their products "from farm to fork" but only from/to their immediate suppliers and immediate customers. If every food facility maintains such records it should be possible to trace the entire food chain.

4 Existing Traceability Schemes

The ability to reduce the costs, both human and financial, in the event of a food recall event depends directly on the ability to locate, or trace, contaminated food products across the food chain. Any traceability solution should address the need by food safety personnel, in the event of a food product recall, to quickly identify companies within their jurisdiction that have a high likelihood of possessing contaminated products. The efficiency and effectiveness of a traceability system depends on the ability to collect, transmit, and analyze information about the handling of food products across all stages of the food chain.

A wide range of traceability schemes are currently in use by food system stakeholders [8, 9, 10, 11, 12, 13]. These systems range from paper-based records to bioactive labeling technology to an array of IT-based solutions from bar codes and radio-frequency identification (RFID) technologies supported by software systems to database management systems. Across the food chain, companies use a variety of these systems which may not be interoperable. An efficient traceability system should be able to link all these different monitoring techniques into an integrated, unified and consistent system.

A necessary requirement to accomplish this integration is the availability of a common standard identification system that is recognized across all stakeholders, or a system to create these translations. Thus, when a contaminated product is confirmed, it would be possible to trace the unique identifier (RFID) or product code (bar code) for that product with all of the companies that were involved in the creation of that food product. In the case of RFID, the tag on the contaminated product would contain the entire history/pedigree for that product. The Global Traceability Standard, a full supply chain traceability solution proposed by the universal standard committee GS1 (General Standard One), recommends the use of Global Location Numbers (GLN), a universal trade unit identification scheme based on the Global Trade Item Number (GTIN), and Electronic Product Codes (EPC) to enable the use of RFID tags to trace products [14, 15]. A methodology for modeling traceability information using the Electronic Product Code Information Service (EPCIS) framework and statecharts in the Unified Modeling Language (UML) to define states and transitions in food product has recently been proposed [16]. While progress has been made in achieving this integration, mostly within large vertically integrated multi-nationals, the difficulties of achieving such a system based on standard codes have been noted above.

In this paper, which extends the work of [17], we explore a different logic-based approach that uses inference rules to determine the set of all companies that may be linked to a contaminated product. Our approach does not depend on the availability of a common standard or unique identifier. Rather, the proposed approach utilizes information about the primary suppliers and customers for all food companies, along with their products—consistent with the “one step forward” and “one step backward” required under FMSA as noted above. In the event of a recall for Product A manufactured by Company X, we use logic programming to compute the set of all companies that are linked to the dyadic unit food-company across the entire supply chain. Using rules, we can trace backward to the set of likely companies that are the possible source of the contamination and can trace forward to identify the destination and location of similarly contaminated products.

We use a form of declarative programming – Answer Set Programming (ASP) [18], to represent complex pathways of the food supply chain and to track-and-trace recalled products and other information of interest to public health officials. ASP has been applied to industrial problems, but to the best of our knowledge it has not been used in food supply chain applications before.

5 ASP Program Encoding

The ASP paradigm is based on the stable models/ answer sets semantics of logic programs [19, 20] and has been shown to be a powerful methodology for knowledge representation, including the representation of defaults, inheritance reasoning, and multiple interesting aspects of reasoning about actions and their effects, as well as being particularly useful to solve difficult search problems. In the ASP methodology, search problems are reduced to the computation of the stable models of the problem. Several ASP solvers—programs that generate the stable models of a given problem encoded in the ASP formalism—have been implemented, e.g. ASSAT, clasp, Cmodels, DLV, GnT, nomore++, Pbmodels, Smodels, etc. In what follows we provide the basic syntactic constructs and the intuitive semantics of the ASP language used in this work. A complete formal specification of the syntax and semantics of the language can be found in [20, 21].

A signature Σ of the language contains constants, predicates, and function symbols. Terms and atoms are formed as is customary in first-order logic. A literal is either an atom (also called a positive literal) or an atom preceded by $\neg$ (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 $lit(\Sigma)$. A rule is a statement of the form:

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

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 $\vee$ corresponds to the disjunction operator. The head of the rule is the part of the statement to the left of symbol $\leftarrow$, while the body of the rule is the part on its right side. Intuitively, the rule meaning is 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 of the rule is substituted by the falsity symbol $\perp$ then the rule is called a constraint. The intuitive meaning of a constraint is that its body must not be satisfied. Rules with variables are used as shorthand for the sets of their ground instantiations. Variables are denoted by capital letters. An ASP program is a pair of $\langle\Sigma, \Pi\rangle$, where Σ is a signature and Π is a set of rules over Σ, but usually the signature is defined implicitly and programs are only denoted by Π. A stable model (or answer set) of a program Π is one of the possible sets of literals of its logical consequences under the stable model/answer set semantics.

Our encoding—the set of rules of program Π—contains roughly 25 rules, while event records (in ASP, rules with an empty body, also called "facts") and the ontologies describing facts, utilized for experiments, are in the thousands. We use the DLV system [22] as our ASP solver.

Advantages of applying the ASP formalism to the food supply chain traceability problem include: (1) ASP can easily encode many forms of domain knowledge, including hierarchical ontologies and heuristics. As shown by some previous works [23, 24], ASP allows generating ontologies for different types of information relevant to this domain, e.g. food, geographical, disease, etc. 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) ASP is well-suited to represent action and change. A food supply chain is an intrinsically dynamic enviroment where food products move from one node, or food operator, to the next node 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 (3) 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 since it constitutes competitive advantage to their business.

5.1 Domain Representation

Given the proprietary nature of supplier/customer base information and the difficulty to obtain this data directly from private sector companies, we turned to data publicly available on the World Wide Web and using web scrapping techniques downloaded and assembled a database of suppliers of food and agricultural products. This database contains more than 6,000 American companies located in all 50 states, the District of Columbia and Puerto Rico, with firms encompassing the whole food supply chain, including: grower, manufacturer, processor, packer, distributor, wholesaler, retailer, etc. Each firm is classified as at least one of these types, but a firm may have more than one role in the supply chain, e.g. it may be a processor and also a wholesaler of its products. Besides the standard information about a firm, i.e. name, address, the database contains a list of the product categories that the firm commercializes, e.g. salad dressing, juice mixed, peanut butter.

We demonstrate the power of using ASP to solve the traceability problem by showing an example involving pork products. For simplicity sake, in this example we assume that the supply chain for pork sausages, shown on Figure 1, encompasses: (a) farmers supplying fresh pork meat to (b) processors supplying chilled or frozen pork to (c) manufacturers of pork sausages supplying (d) wholesalers of pork sausages supplying (e) retailers who sell pork sausages to consumers. A small number of companies that populate this supply chain, as identified in our assembled supplier database, are also shown in Figure 1 in the form of a directed graph. In this graph, each node corresponds to a company identified by an id code, and an edge originating from a company/node A and connecting it to a company/node B expresses a supplier-customer relationship where A supplies certain food product to B. In addition, each type of company/ node aligns vertically with its role or category in the pork supply chain represented at the top of Figure 1. For example, company "cp3092" corresponds to a farmer who supplies fresh pork meat to three processors identified by codes "cp123", "cp393", and "cp684"; processor "cp123" supplies chilled or frozen pork meat to four manufacturers, e.g. "cp273"; and so forth. In the ASP knowledge base, each company is modeled by three types of "facts," rules (2)-(4).

```
company(Idcode,Name,State).                                   (2)
type_company(Idcode,Type).                                    (3)
prod_supplied(Idcode,Product).                                (4)
```

In our model, for the purpose of this example, each company is represented by a single rule (2), which identifies it by an id code, its name, and the state where the company is located. For simplification, we assume that each company has a single facility and this is the state where the supplied product originates and is shipped to others. Rules of type (3) indicate the role each company exerts in the supply chain. As mentioned before, it is not uncommon that a given company may have more than one role, e.g. a wholesaler may also be a retailer who sells directly to consumers. Thus, such company will have at least two rules of type (3), one to indicate that the company is of type "wholesaler" and the other that the company is a "retailer". It is very common for a given company to supply several products, and thus, our knowledge base contains a rule of type (4) for each of these products. Once a recall of a product sold by a certain company is issued, this information is added to the knowledge base in the form of rule (5), with the company being identified by the id code.

```
recall(Product,Idcode).                                       (5)
```

The expected course of action at this point is that the contaminated product, and its derivative products, are taken out of the market and destroyed. Since only limited information is made available to food safety officials about which companies may be affected—those who received the tainted product or supplied a related contaminated product—delays in the recall process put in risk human lives. Our approach works to reduce these latencies by generating all possible paths this product may have travelled through the supply chain graph. We generate each complete path, from farmer to retailer, for the product in question, as described in the next section.

First, assume that wholesaler company "cp1050" recalls their "porksausage" product. 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. At each stage of the supply chain the product supplied from a company A becomes an ingredient to the company B to which it has been supplied. In the case of pork sausages, the ontology contains facts (6)-(11) which express the production process sequence for pork products illustrated on Figure 1.

```
is_ingr(porkfresh,porkchilled).                               (6)
is_ingr(porkchilled,porkfrozen).                              (7)
is_ingr(porkchilled,porksausage).                             (8)
is_ingr(porkfresh,porkfrozen).                                (9)
is_ingr(porkfrozen,porksausage).                              (10)
is_ingr(porksausage,porksausage).                             (11)
```

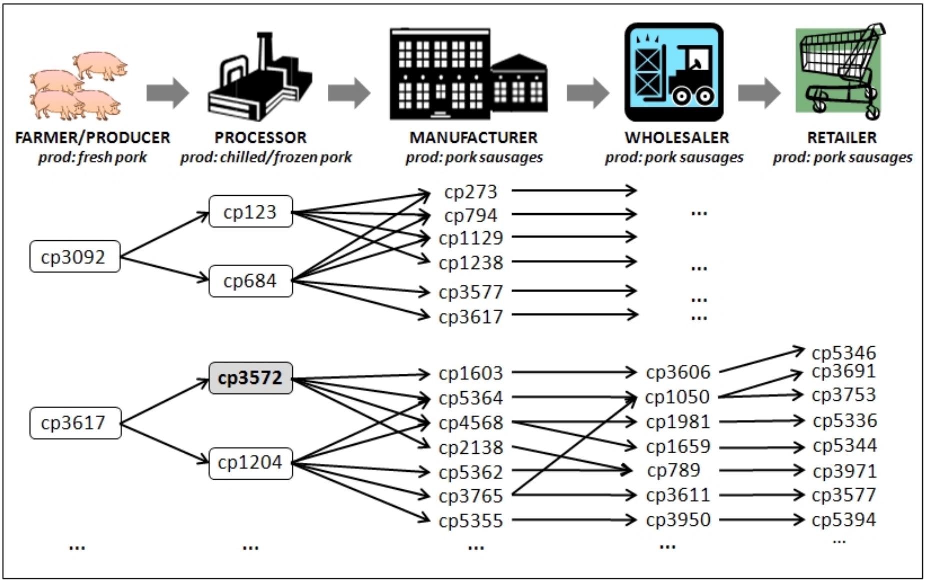

Fig. 1. Illustrative Supply Chain for Pork Sausages

5.2 Generating Supply Chain Paths

We use a two-step approach to solve the problem of identifying companies affected by a food recall when incomplete information may hinder the process and create delays. In the first step, we generate all supply chain paths for pork products with rules of type (12)-(17), where the supplied pork product at each level of the supply chain is used to prune the search among all other possible combinations of food products represented in our knowledge base. Intuitively, (12) means that a five-tuple *supply_chain(G,P,M,W,R)* represents the complete path of production of a given final product, e.g. pork sausages, from grower/producer to processor to manufacturer to wholesaler to retailer. Rules (13)-(17) compute the individual supplier-customer relations, or edges of the supply chain graph.

```
supply_chain(G,P,M,W,R) :-
  produces(G,porkfresh),processes(P,porkchilled),
  manufactures(M,porksausage),wholesells(W,porksausage),
  sells(R,porksausage).                                    (12)
```

```
produces(C,F) :-
  company(C,_,_),type_company(C,grower),                   (13)
  prod_supplied(C,F), F==porkfresh.
```

```
processes(C,F) :-
  company(C,_,_),type_company(C,processor),                (14)
  prod_supplied(C,F), F==porkchilled.
```

```
manufactures(C,F) :-
  company(C,_,_),type_company(C manufacturer),          (15)
  prod_supplied(C,F), F==porksausage.
```

```
wholesells(C,F) :-
  company(C,_,_),type_company(C,wholesaler),             (16)
  prod_supplied(C,F), F==porksausage.
```

```
sells(C,F) :-
  company(C,_,_),type_company(C,retailer),               (17)
  prod_supplied(C,F), F==porksausage.
```

In the second step, each such supply chain path is broken down and ex-pressed as individual supplier-customer relations by rules (18)-(21). The reason for converting the supply chain back to these relations is to improve the efficiency of the computation during the tracing stage. Rule (18), and similarly rules (19)-(21), intuitively expresses that a grower / producer company G supplies fresh pork to a processor company P which utilizes this product as the main ingredient to produce and supply chilled pork to its customers. Rule (18), as well as (19)-(21), also enforces that companies G and P are not the same to ensure that the supply chain graph is cycle free.

```
supplies(G,porkfresh,P) :-
  supply_chain(G,P,M,W,R),company(G,_,_),
  type_company(G,grower),prod_supplied(G,porkfresh),     (18)
  company(P,_,_),type_company(P,processor),
  prod_supplied(P,porkchilled), G!=P.
```

```
supplies(P,porkchilled,M) :-
  supply_chain(G,P,M,W,R),company(P,_,_),
  type_company(P,processor),prod_supplied(P,porkchilled), (19)
  company(M,_,_),type_company(M,manufacturer),
  prod_supplied(M,porksausage), P!=M.
```

```
supplies(M,porksausage,W) :-
  supply_chain(G,P,M,W,R),company(M,_,_),
  type_company(M,manufacturer),
  prod_supplied(M,porksausage),                          (20)
  company(W,_,_),type_company(W,wholesaler),
  prod_supplied(W,porksausage), M!=W.
```

```
supplies(W,porksausage,R) :-
  supply_chain(G,P,M,W,R),company(W,_,_),
  type_company(W,wholesaler),                            (21)
  prod_supplied(W,porksausage),
  company(R,_,_),type_company(R,retailer),
  prod_supplied(R,porksausage), W!=R.
```

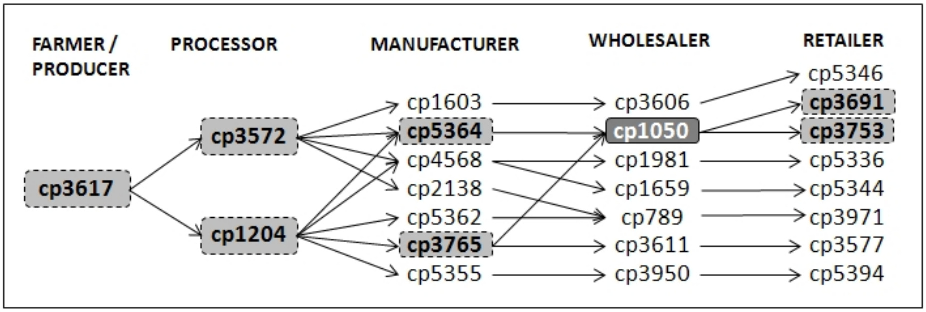

Fig. 2. Tracing Contaminated Pork Sausages in the Supply Chain

5.3 Tracing Contaminated Products

The goal of tracing the contamination forward in the supply chain from the point of recall, e.g. wholesaler "cp1050," is achieved by rules (22) and (23). Rule (22) says that if recalling company C, located in state LC, supplies its recalled food product F to company A, located in state LA, then LA may be affected by the recall and is part of the contamination. Thus, company A must be inspected by food safety officials to verify that its entire contaminated product is taken out of the market. Rule (23) propagates this trace to the next forward stage of the supply chain. Figure 2 shows an example of firms affected by a recall after tracing back and forward in the supply chain such product. Similarly, rules (24) and (25) trace back the contaminated product through the supply chain.

```
forward_trace(C,LC,F,A,LA) :-
  recall(F,C),supplies(C,F,A),                              (22)
  company(C,_,LC),company(A,_,LA),C!=A.
```

```
forward_trace(B,LB,F1,A,LA) :-
  company(B,_,LB),company(A,_,LA),company(C,_,LC),          (23)
  supplies(B,F1,A),is_ingr(F,F1),
  forward_trace(C,LC,F,B,LB), B!=C, B!=A, A!=C.
```

```
backward_trace(A,LA,F,C,LC) :-
  recall(F,C),supplies(A,F,C),                              (24)
  company(C,_,LC),company(A,_,LA),C!=A.
```

```
backward_trace(B,LB,F1,C,LC) :-
  company(B,_,LB),company(A,_,LA),company(C,_,LC),          (25)
  supplies(B,F1,C),is_ingr(F1,F),
  backward_trace(C,LC,F,A,LA),B!=C, B!=A, A!=C.
```

Finally, when these rules are submitted to the answer set solver DLV, we obtained the following list of atoms which corresponds to the solution of the traceability problem illustrated in Figure 2. In addition to the rules listed above a couple of other rules are used to retrieve the name of the recalling company, companies supplied the contaminated product directly, and their downstream customers. These companies are named in atoms of the type *affected_comp(Idcode, Name, State)*. Company names and codes appearing in this example are for illustrative purposes only and do not correspond to real company names in the knowledge base. Note that using ASP we can further focus the search and obtain a list of affected companies on a given state.

```
{recalling_comp(cp1050,atrading,ca),
  forward_trace(cp1050,ca,porksausage,cp3691,il),
  forward_trace(cp1050,ca,porksausage,cp3753,il),
  affected_comp(cp3691,gustopack,il),
  affected_comp(cp3753,apacking,il),
  backward_trace(cp3617,il,porkfresh,cp3572,il),
  backward_trace(cp3617,il,porkfresh,cp1204,ca),
  backward_trace(cp3572,il,porkchilled,cp5364,il),
  backward_trace(cp1204,ca,porkchilled,cp3765,il),
  backward_trace(cp5364,il,porksausage,cp1050,ca),
  backward_trace(cp3765,il,porksausage,cp1050,ca).}
```

Assume now that processor firm "cp3572" is recalling its chilled pork product. To find a solution to this new contamination problem one needs only to add new fact (26). The list of atoms below shows a portion of the results computed by DLV.

```
recall(porkchilled,cp3572).
{recalling_comp(cp3572,ainc,il),
  forward_trace(cp3572,il, porkchilled,cp5364,il),
  forward_trace(cp3572,il, porkchilled,cp4568,la),
  forward_trace(cp3572,il, porkchilled,cp2138,wi),
  forward_trace(cp3572,il, porkchilled,cp1603,ok),
  forward_trace(cp5364,il,porksausage,cp1050,ca),
  forward_trace(cp4568,la,porksausage,cp1981,wi),
  forward_trace(cp4568,la,porksausage,cp1659,co),
  forward_trace(cp2138,wi,porksausage,cp789,fl),              (26)
  forward_trace(cp1603,ok,porksausage,cp3606,il),
  forward_trace(cp1050,ca,porksausage,cp3691,il),
  forward_trace(cp1050,ca,porksausage,cp3753,il),
  forward_trace(cp1981,wi,porksausage,cp5336,ca),
  forward_trace(cp789,fl,porksausage,cp3971,il),
  forward_trace(cp3606,il,porksausage,cp5346,ga),
  forward_trace(cp1659,co,porksausage,cp5344,ny),
  backward_trace(cp3617,il,pork-fresh,cp3572,il),
  affected_comp(cp1603,afoods,ok),
  affected_comp(cp1050,atrading,ca), ...}
```

6 Complex Supply Chains

Simple and linear supply chains such as the pork example depicted in Figure 1 are not common. Increased market forces, as trade and globalization of products, have added high complexity to supply chains. Thus, the solution implementation discussed on the previous sections may not be as useful or adequate.

A more realistic and complex supply chain for peanut products is presented in Figure 3. Similar to the pork example, this supply chain includes a linear trajectory of transformation of a commodity, i.e. raw peanuts, into a final product, i.e. peanut butter. This (national) supply chain moving in a straight line from farmer/producer to retailer is shown at the top portion of Figure 3. In addition, several other sup-ply chain pathways are presented at the bottom of Figure 3 including export outside the U.S. It includes other stakeholders such as (a) brokers who market or sell raw peanuts supplied by wholesalers, blanched peanuts supplied from processors, and/or peanut butter supplied from manufacturers to (b) distributors supplying these products, and others commercialized directly from manufacturers, to national wholesalers and to (c) exporters supplying products to international distributors selling both commodities and final products through the supply chain of another country.

Such more complex supply chains allow a better demonstration of the expressive power of ASP. Even though now the supply chain has a larger number of ramifications and new stakeholders, modeling the additional elements is straightforward. Each firm belonging to the peanut supply chain in our supplier database is still modeled in the ASP knowledge base by rules (2)-(4). Three new types of companies are included: “distributor”, “broker,” and “exporter”. As before, some companies may have more than one role in the supply chain, e.g. a broker company may also be a distributor of products, and thus, those will be modeled by as many rules of type (3) as the roles it performs. Similarly, to the pork example, companies may commercialize several products which will be modeled by as many rules of type (4).

The production stages of the supply chain, where raw peanuts are supplied to a processor to be transformed into prepared and preserved peanuts used to produce crude peanut oil, or transformed into blanched peanuts, then supplied to a manufacturer of peanut oil or peanut butter, are captured by (27)- (31) in food our ontology.

```
is_ingr(peanutraw,pnprepared).                    (27)
is_ingr(pnprepared,pncrudeoil).                   (28)
is_ingr(peanutraw,pnblanched).                    (29)
is_ingr(pnblanched,pnbutter).                     (30)
is_ingr(pncrudeoil,peanutoil).                    (31)
```

6.1 Modeling Complex Supply Chains

Generating all possible production pathways for the peanut supply chain using the previous implementation is possible, but cumbersome. It is clear that this approach would be highly inefficient when encoding larger and more complex supply chains.

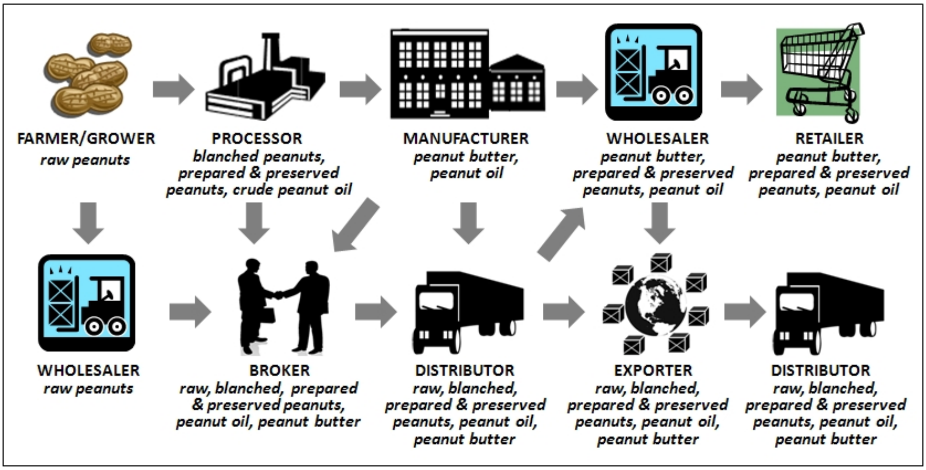

Fig. 3. More Complex Example of Supply Chain for Peanut Products

Rules (13)-(17) are also insufficient to express all supplying relationships among stakeholders of this example and additional rules are required. As shown in Section 5.2, encoding each supplier-customer relation as a rule is more efficient and corresponds to modeling each arrow appearing in the supply chain illustration of Figure 3. Thus, we substitute rules (18)-(21) by new rules (32)-(45).

Rule (32) is the "de facto" generator of all supplier-customer relations. Intuitively it is read as: company A supplies ingredient and/or product I1 to company B if, and only if, A is a valid supplier of subproducts of food F to B, where I1 (supplied by A) and I2 (supplied by B) are both subproducts of F.

```
supplies(A,I1,B) :-
  is_of(I1,F),is_of(I2,F),valid_supplier(F,A,B),          (32)
  prod_supplied(A,I1),prod_supplied(B,I2).
```

By using a single, generic rule (32) to generate all supplying relations, we allow for a more general modeling of all valid supplier-customer connections, further encoded by rules with head predicate *valid_supplier(F,A,B)*, or (33)-(45). Rules (33)-(36) encode the linear supply chain path from grower to retailer, similarly to (18)-(21) in the pork example, while (37)-(45) represent supply relations to the new types of stakeholders appearing in the peanuts example.

Rules (33)-(34) model the transformative stages of the supply chain and express that a company A is a valid supplier, of a product made with food F, to a company B if the type of roles A and B have in the supply chain of F are such that product I1 supplied by A to B is the ingredient for product I2 supplied by B to other firms.

```
valid_supplier(F,A,B) :-
  supply_chain(F),is_of(I1,F),is_of(I2,F),is_ingr(I1,I2),
  type_company(A,grower),prod_supplied(A,I1),                    (33)
  type_company(B,processor),  prod_supplied(B,I2),A!=B.
```

```
valid_supplier(F,A,B) :-
  supply_chain(F),is_of(I1,F),  is_of(I2,F),is_ingr(I1,I2),     (34)
  type_company(A,processor),  prod_supplied(A,I1),
  type_company(B,manufacturer),prod_supplied(B,I2),A!=B.
```

Since the other supplying relations to be modeled consist of customer companies commercializing the product supplied by the supplier companies, i.e. no food transformation process occurs, we add one rule to represent each commercialization step, similar to rules (35)-(38). Thus, overall a total of 11 such rules, i.e. (35)-(45), are added. The meaning of these rules is that, given companies A and B, A is a valid supplier of a product I made with food F to B, if the roles of A and B in the supply chain of F are such that B supplies the same product I to other firms. These rules can also be used for the pork example since, in the absence of such supplier-customer relations, rules (35)-(45) will not be fired and that example will produced the expected results.

```
valid_supplier(F,A,B) :-
  supply_chain(F),is_of(I,F),
  type_company(A,manufacturer),prod_supplied(A,I),               (35)
  type_company(B,wholesaler),prod_supplied(B,I),A!=B.
```

```
valid_supplier(F,A,B) :-
  supply_chain(F),is_of(I,F),                                    (36)
  type_company(A,wholesaler),prod_supplied(A,I),
  type_company(B,retailer),prod_supplied(B,I),A!=B.
```

```
valid_supplier(F,A,B) :-
  supply_chain(F),is_of(I,F),                                    (37)
  type_company(A,grower),prod_supplied(A,I),
  type_company(B,wholesaler),prod_supplied(B,I),A!=B.
```

During a food recall, a supply chain of a particular food product can now be targeted, among a number of existing food chains, simply be adding a fact of type *supply_chain(S)* to the ASP knowledge base. Fact (46) means that the only supplying relationships to be proven by the ASP solver are those belonging to the peanut supply chain. For the pork example, we would add a similar fact where *S=pork*.

```
supply_chain(peanut).                                            (46)
```

In addition to (27)-(31), raw ingredients and other subproducts of peanuts—the food product of this supply chain—are modeled in the ASP knowledge base by facts (47)-(52). They express that certain raw ingredients or subproducts, i.e. raw peanuts, crude peanut oil, are part of the peanut supply chain, since they are made from peanuts. New facts of type *is_of(Prod,S)* would also be added for the pork example.

```
is_of(peanutraw,peanut).                                  (47)
is_of(pnprepared,peanut).                                 (48)
is_of(pncrudeoil,peanut).                                 (49)
is_of(pnblanched,peanut).                                 (50)
is_of(peanutoil,peanut).                                  (51)
is_of(pnbutter,peanut).                                   (52)
```

Alternatively, facts describing that a product I1 is an ingredient of a product I2, i.e. facts of type *is_ingr(I1,I2)*, could be combined with the facts associating each ingredient/product I to a specific food supply chain S, i.e. *is_of(I,S)*, to form a new predicate that would substitute those: *is_ingr_of(I1,I2,S)*. This would provide a more concise encoding, especially for foods with a large number of subproducts, but still be less general than a more developed ontology for describing this supply chain's food.

The use of the complete peanut ontology would have "peanut" at its root node and thus, the ASP program would entail that products I1 and I2 are in fact subproducts of peanut. Thus, the root node of each food hierarchy would characterize each corresponding supply chain. As we model only a very small portion of the peanut hierarchy in this example, we have opted to present a simpler, but less elegant, representation in this program. To include a complete ontology for peanut would require additional rules and the partial modification of the ASP program rules.

6.2 Tracing Contamination in Complex Supply Chains

Rules (22)-(23) for tracing forward from the point of recall require no modification and are included in the new program as rules (53)-(54). As in the pork example, (54) ensures that the ASP program will recursively trace forward companies affected by the recall which are (a) more than one step removed from the company recalling the contaminated product and (b) use product F received from a supplier as an ingredient to produce their product F1. However, since it is valid for some firms to re-sell products, as expressed by (35)-(45), we need to add recursive rule (55) to capture the situation where both a supplier and its customer supply the same product F. Similarly, contaminated products are traced backward in the supply chain by rules (56)-(58). Rules (24)-(25) used in the pork example are renamed as (56)-(57), and (58) is a new rule added to encode cases of supplier and customer supplying a same product.

```
forward_trace(B,LB,F,A,LA):-
  supplies(B,F,A),forward_trace(C,LC,F,B,LB),
  company(B,_,LB),company(A,_,LA),company(C,_,LC),       (55)
  B!=C, B!=A, A!=C.

backward_trace(B,LB,F,C,LC) :-
  supplies(B,F,C), backward_trace(C,LC,F,A,LA),
  company(B,_,LB),company(C,_,LC),company(A,_,LA),       (58)
  B!=C, B!=A, A!=C.
```

While the above encoding has been developed for a complex peanut supply chain it can easily accommodate the previous pork example simply by including facts of type *is_of(I,S)* for pork. ASP programs that require only small changes to accommodate new or changing circumstances, as this one does, are called "elaboration tolerant". Programs exhibiting such property are highly preferable.

7 Conclusions

This paper demonstrates the utility of answer set programming in identifying not only the source of a food contamination but also the location of contaminated products across complex food chains for pork and peanut products. We represent all possible paths of a contaminated product across the supply chain as a sequence of stages by which a food product 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. Using rules of inference, we then reduce the set of all possible pathways of contamination based on information contained in the recall. We are also able to capture the process by which contaminated products become ingredients in other products during sequential stages of production. The logic-based approach developed herein is well-suited to be used by state agencies charged with inspecting food production, distribution and retail facilities in the event of a national recall. The approach is particularly useful for ingredient-driven contaminations in which the contaminated product is used as an ingredient in a broad set of secondary products.

References

1. Scallan, E., Hoekstra, R.M., Angulo, F.J., Tauxe, R.V., Widdowson, M.-A., Roy, S.L., Jones, J.L., Griffin, P.M.: Foodborne Illness Acquired in the United States—Major Pathogens. Emerging Infectious Diseases 17(1), 7–15 (2011)
2. Scallan, E., Griffin, P.M., Angulo, F.J., Tauxe, R.V., Hoekstra, R.M.: Foodborne Illness Acquired in the United States—Unspecified Agents. Emerging Infectious Diseases 17(1), 16–22 (2011)
3. Scharff, R.L.: Health-Related Costs From Foodborne Illness in the United States. The Produce Food Safety Project at Georgetown University (2010)
4. Fritz, M., Schiefer, G.: Tracking, tracing and business process interests in food commodities: A multi-level decision complexity. Int'l Journal Production Economics 117(2), 317–329 (2009)
5. U.S. Department of Health and Human Services: Office of Inspector General. Traceability in the Food Supply Chain, Report OEI-02-06-00210 (March 2009)
6. FDA Food Safety Modernization Act of 2010, S. 510, 111th Congress, 2nd Session. Signed into law by President Barack Obama (January 4, 2011)
7. U.S. Food and Drug Administration. Pilot Projects for Improving Product Tracing along the Food Supply System (September 2011)
8. Buhr, B.L.: Traceability and information technology in the meat supply chain: implications for firm organization and market structure. Journal of Food Distribution Research 34(3), 13–26 (2003)

9. Raschke, A., Strich, S., Huppke, S., Neugebauer, M., Geuther, E., Bertling, W., Walders, B., Reiser, C., Hess, J.: Induction and detection of long-lasting peptide-specific antibody responses in pigs and beef cattle: A powerful technology for tracing meat processing chains from stock farmers to sales counters. Food Control 17(1), 65–74 (2006)
10. Regattieri, A., Gamberi, M., Manzini, R.: Traceability of food products: general framework and experimental evidence. Journal of Food Engineering 81, 347–356 (2007)
11. Bulut, H., Lawrence, J.D.: Meat Slaughter and Processing Plants' Trace-ability Levels: Evidence From Iowa. Working Paper #08015, Working Paper Series, Department of Economics, Iowa State University (April 2008)
12. Shanahan, C., Kernan, B., Ayalew, G., McDonnell, K., Butler, F., Ward, S.: A framework for beef traceability from farm to slaughter using global standards: an Irish perspective. Computers and Electronics in Agriculture 66(1), 62–69 (2009)
13. Souza-Monteiro, D.M., Caswell, J.A.: Traceability adoption at the farm level: An empirical analysis of the Portuguese pear industry. Food Policy 34(1), 94–101 (2009)
14. Fritz, M., Schiefer, G.: Tracking and tracing in food networks. In: World Conference on Agricultural Information and IT (IAALD AFITA WCCA 2008), August 24-27, pp. 967–972. Tokyo University of Agriculture, Tokyo (2008)
15. GS1. Business Process and System Requirements for Full Supply Chain Traceability. GS1 Global Trace-ability Standard, Issue 1.2.2 (March 2010)
16. Thakur, M., Sørensen, C.-F., Bjørnson, F.O., Forås, E., Hurburgh, C.R.: Managing food traceability information using EPCIS framework. Journal of Food Engineering 103(4), 417–433 (2011)
17. Nogueira, M.L., Greis, N.P.: Recall-driven Product Tracing and Supply Chain Tracking using Answer Set Programming. In: Filipe, J., Dietz, J.L.G. (eds.) Proc. 4th Int'l Conf. Knowledge Eng. and Ontology Development (KEOD 2012), pp. 125–133. SciTePress (2012)
18. Marek, V.W., Truszczynski, M.: Stable models and an alternative logic programming paradigm. In: The Logic Programming Paradigm: A 25-Year Perspective, pp. 375–398. Springer, Berlin (1999)
19. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Kowalski, R., Bowen, K. (eds.) International Logic Programming Conference and Symposium, pp. 1070–1080. MIT Press (1988)
20. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and dis-junctive databases. New Generation Computing 9, 365–385 (1991)
21. Niemela, I., Simons, P.: Extending the Smodels System with Cardinality and Weight Constraints. In: Logic-Based Artificial Intelligence, pp. 491–521. Kluwer Academic Publishers (2000)
22. Leone, N., et al.: The DLV system. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) JELIA 2002. LNCS (LNAI), vol. 2424, pp. 537–540. Springer, Heidelberg (2002)
23. Nogueira, M.L., Greis, N.P.: Rule-Based Complex Event Processing for Food Safety and Public Health. In: Bassiliades, N., Governatori, G., Paschke, A. (eds.) RuleML 2011 - Europe. LNCS, vol. 6826, pp. 376–383. Springer, Heidelberg (2011)
24. Nogueira, M.L., Greis, N.P.: Application of Answer Set Programming for Public Health Data Integration and Analysis. In: Tjoa, A.M., Quirchmayr, G., You, I., Xu, L. (eds.) ARES 2011. LNCS, vol. 6908, pp. 118–134. Springer, Heidelberg (2011)