A computational logic-based representation of the WHO and UNICEF estimates of national immunization coverage.

Burton A, Gacic-Dobo M, Karimov R, Kowalski R

24 January 2011

Anthony BURTON, Department of Immunization, Vaccines and Biologicals, World Health Organization, Avenue Appia 20, 1211 Geneva 27, Switzerland.

Marta GACIC-DOBO, Department of Immunization, Vaccines and Biologicals, World Health Organization, Avenue Appia 20, 1211 Geneva 27, Switzerland.

Rouslan KARIMOV, Division of Policy and Practice, United Nations Children's Fund, 3 United Nations Plaza, New York, NY 10012, USA.

Robert KOWALSKI, Department of Computing, Imperial College London, 180 Queen's Gate, London SW7 2BZ, United Kingdom.

Abstract

Production of official statistics frequently requires expert judgement to evaluate and reconcile data of unknown and varying quality from multiple and potentially conflicting sources. Shocks to the system and exceptional events may be difficult to incorporate in modelled estimates. Computational logic provides a methodology and tools for incorporating analyst's judgement, integrating multiple data sources and modelling methods, ensuring transparency and replicability, and making documentation computationally accessible. Representations using computational logic can be implemented in a variety of computer-based languages for automated production. Computational logic complements standard mathematical and statistical techniques and extends the flexibility of formal modelling.

A basic overview of computational logic is presented and its application to official statistics is illustrated with the WHO & UNICEF estimates of national immunization coverage.

Key words: official statistics, knowledge representation and reasoning, artificial intelligence, computational logic, monitoring, health service statistics, global health

1. Introduction

Official statistics, particularly at the international level[1,2,3], often rely on potentially conflicting data of unknown and varying quality from multiple sources; expert judgement is frequently required to evaluate and reconcile these data. Shocks to the system and deviation from general trends and patterns may be difficult to incorporate in succinct models. Current methods for ensuring transparency, replicability, and sufficiently detailed documentation in these circumstances is challenging, frequently inadequate and cumbersome.

Computational logic[4,5], a form of symbolic logic developed in artificial intelligence, provides a powerful and flexible methodology and set of tools that is especially well-suited for formally describing complex situations. Models described in computational logic can also take advantages of computer-based languages for large scale implementation.

Since 2000 the World Health Organization (WHO) and the United Nations Children's Fund (UNICEF) have made annual estimates of national infant immunization coverage for selected vaccines[6]. Estimates are based on reports to WHO and UNICEF submitted by national authorities and are supplemented with results from nationally representative household or community surveys. Local staff, primarily national immunization system managers and WHO/UNICEF regional and national staff, are consulted for information on the performance of specific immunization systems and factors that might influence or

bias empirical data. Estimates are derived through a country-by-country review of available data informed and constrained by a set of heuristics - some of which are described below - and make only limited use of statistical and mathematical models[6]. While the final estimates may not differ from data reported by national authorities, they constitute an independent technical assessment by WHO and UNICEF of the national immunization system performance. Annual country-specific estimates from 1980 are available at: http://www.who.int/immunization_monitoring/en/globalsummary/wucoveragecountrylist.cfm and http://www.childinfo.org/immunization_countryreports.html. Additional analyses can be found at: http://www.who.int/immunization_monitoring/data/en/ and http://www.childinfo.org/immunization.html.

Previously, the informal articulation and manual application of the estimation procedure has led, in some instances, to inconsistent estimates (not all estimates adhere to the heuristics), irreproducible results and to insufficiently informative accompanying documentation. To address these issues and improve the transparency of the methods, computational logic has been used to formally represent the rules, data and decisions, from which the WHO and UNICEF estimates of national immunization coverage (WUENIC) may be logically inferred.

The declarative nature of the formalization lends itself to a fairly direct translation to logic programming and expert-system computer languages. To take advantage of automated data processing the formal representation has been implemented in the general-purpose logic programming language Prolog[7] which implements computational logic. The implementation was first used in May 2010 to support the production of estimates for the period 1997-2009. The formal representation and Prolog code are available at: ***http://....***

2. Knowledge representation and reasoning

Computational logic which is both simpler and more powerful than conventional symbolic logic is used to represent knowledge (and assumptions) and to derive logical consequences of that knowledge. Knowledge represented in computational logic can be viewed as a relational database extended by rules expressed in logical form. Such representations are often called “knowledge bases”.

In computational logic, logical consequences of information in a knowledge base are derived by means of an inference engine, which implements a mechanical reasoning procedure.

In our application domain-specific knowledge of immunization coverage is represented in computational logic and the inference engine derives estimates of immunization coverage. The knowledge consists of:

1. *data* and other domain-specific information relevant to immunization coverage. The data include coverage reported by national authorities and results from national household or community surveys. Other information includes knowledge about the quality and relevance of reported data and surveys (e.g. survey sample size), assessments of national monitoring systems and the occurrence of programmatic and exogenous factors influencing immunization system performance (e.g., vaccine supply shortages, changes in immunization policies, civil unrest);
2. *rules* representing the policies and procedures used to derive estimates from the data and information, to define domain-specific concepts, and perform computations.
3. *decisions* made by the working group both to override and to augment the rules. Such decisions are explicitly identified and are accompanied with an explanation.

The data, rules and decisions are represented in computational logic by means of two simple kinds of sentences: *atomic sentences* (also called *facts*), which have no subparts that are also sentences and *conditionals* , which have the form ***if*** *condition(s)* ***then*** *conclusion* or equivalently, *conclusion* ***if*** *condition(s)*. Such conditionals (also called *implications*) combine an atomic *conclusion* with a conjunction of *conditions*[8].

In the remainder of this section, the logic-based approach is presented and illustrated with simplified examples taken from our application.

2.1 Facts (or atomic sentences)

*Atomic sentences (*or *facts)* consist of a predicate (or relationship) with a number of arguments (or parameters). In symbolic notation, facts are written with the predicate first, followed by the arguments, separated by commas and surrounded by parentheses. For example, *data reported by national authorities* is represented as:

reported(*country, vaccine, year, coverage*)

where **reported** is a predicate and ***country, vaccine, year, coverage*** are the arguments of the predicate. The ***coverage*** represents the proportion of children below one year of age in the ***country*** vaccinated during the ***year*** with the ***vaccine***, as reported by the national authorities.

For example, the fact that coverage for the third dose of diphtheria, tetanus and pertussis vaccine (DTP3) in 2004 reported by the Egyptian national authorities was 97% is represented as:

reported(egy, dtp3, 2004, 97)

Survey results are represented in the form:

survey(*coutry, vaccine, year, coverage*)

For example, the fact that a Demographic and Health Survey (DHS) found 93.5% DTP3 coverage in Egypt for a sample of children born in 2004 is represented as:

survey(egy, dtp3, 2004, 93.5)

In relational databases, predicates are relations, which can be viewed as tables. For example, the **reported** and **survey** predicates could be pictured as separate tables:

reported data

country	***vaccine***	***year***	***coverage***
egy	dtp3	2004	97
egy	dtp3	2005	96
…….	…..	…..	……..

survey data

country	***vaccine***	***year***	***coverage***
egy	dtp3	2004	93.5
egy	dtp3	2005	95
…….	…..	…..	……..

Each row in the table corresponds to an atomic sentence in logic. The table name corresponds to the predicate of the sentence, and each column corresponds to an argument of the predicate.

The **reported** and **survey** predicates record the basic input from which the estimates of immunization coverage are derived as output. The next section describes how the output is derived by applying domain-specific rules to the input. The estimate (output) is represented using the predicate and arguments:

wuenic(*country, vaccine, year, coverage*)

This output can also be represented as a table:

wuenic

country	***vaccine***	***year***	***coverage***
egy	dtp3	2004	97
egy	dtp3	2005	96
…			

In developing a logic-based representation it is necessary to decide on the choice of predicates and arguments. This corresponds to the decision regarding the choice of relations (or tables) in a relational database. Frequently many alternative representations are possible, and similar considerations apply in both cases. For example, an alternative representation is to employ a single predicate:

data(*source, country, vaccine, year, coverage*)

corresponding to a single table:

data

source	***country***	***vaccine***	***year***	***coverage***
reported	egy	dtp3	2004	97
reported	egy	dtp3	2005	96
survey	egy	dtp3	2004	93.5
survey	egy	dtp3	2005	95
wuenic	egy	dtp3	2004	97
wuenic	egy	dtp3	2005	96
……..	…….	…..	…..	…….

2.2 Rules (or conditionals)

The estimates are derived from the data using domain-specific rules[6] expressed as logical conditionals. The domain specific rules can be expressed in symbolic form, which facilitates their computer-based implementation but they can also be expressed in informal natural languages (e.g., English, French). For example the rule that derives the output estimate from the input data when there are both reported data and survey results in the same year and the two data values are within 10% of one another can be expressed informally as the English language rule:

> If, for a given country/vaccine/year, the reported data are within 10% points of the survey results, then the estimate is the reported data.

As an intermediate representation, between informal English and the symbolic form, the same rule can also be expressed in more precise English:

For every *country C, vaccine V, year Y, reported coverage* P_{rpt} *and survey coverage* P_{surv}*,*

If *the coverage in country C, vaccine V, and year Y is reported by*

the national authorities as P_{rpt}

and survey coverage result for country C, vaccine V and year Y is P_{surv}

and the absolute difference between P_{surv} *and* P_{rpt} *is less than 10*

then *the estimate for country C, vaccine V and year Y is* P_{rpt}*.*

In symbolic notation of the form of computation logic used in this application the rule above is written in the *conclusion **if** conditions* form:

wuenic (*C, V, Y, Prpt*) :-
reported(*C, V, Y, Prpt*),

survey(*C, V, Y, Psurv*),
abs(*Psurv - Prpt*) < 10.

Here *C,V,Y,Prpt$_t$, Psurv* are variables standing for any country, vaccine, year, reported coverage and survey coverage respectively. The variables are said to be universally quantified. In general variables are represented by expressions beginning with an uppercase character, "and" is represented by a comma, and "if" is represented by ":-".

The *conclusion* of a rule (or conditional) is an *atomic expression,* which is like a fact, consisting of a predicate and its arguments, but, unlike a fact, may contain variables. The *conditions* are a conjunction of atomic expressions or negations of atomic expressions which may also contain variables.

A rule containing universally quantified variables stands for all variable-free instances of the rule. For example, the rule above logically implies the variable-free instance

wuenic(egy, dtp3, 2004, 97) :-
reported(egy, dtp3, 2004, 97),
survey(egy, dtp3, 2004,93.5),
abs(93.5 - 97) < 10.

The inference engine applies the rule to the atomic sentences representing the basic data using a definition of the arithmetic function **abs** and the relation "<" to derive the estimate:

wuenic(egy, dtp3, 2004, 97).

2.3 Quantitative computation

Quantitative calculations and procedures can also be implemented in computational logic. For example, an estimate of coverage for the first dose of DTP can be made based on a second degree polynomial function with parameters estimated by a modelled relationship between DTP1 and DTP3 survey results[6].

wuenic (*C*, dtp1, *Y, Pdtp1*) :-
wuenic(*C*, dtp3, *Y, Pdtp3*),
***Pdtp1* is *Pdtp3* + (-0.0066 * *(Pdtp3 * Pdtp3)) + (*0.4799 * *Pdtp3)* +**
16.67.

Linear interpolation of a value between two other values may be implemented as:

interpolate(*Yearbefore, Pbefore, Yearafter, Pafter, Yearinter, Pinter*) :-
***Pinter* is *Pbefore* +**

(Yearinter - Yearbefore) * ((Pafter - Pbefore) / (Yearafter - Yearbefore)).

Interpolation is used, for example, to estimate missing data between two years of reported data.

In both of these examples "is" is an auxiliary predicate representing equality.

2.4 Auxiliary predicates

In addition to the input predicates, such as **reported** and **survey**, calculations, and the output predicate **wuenic**, our application uses pre-defined functions and predicates, such as "**abs",** "<", "is". Special purpose, more abstract auxiliary predicates may be defined and used to express more general rules. For example, the earlier rule:

wuenic (*C, V, Y, Prpt*) :-
reported(*C, V, Y, Prpt*),
survey(*C, V, Y, Psurv*),
abs(*Psurv - Prpt*) < 10.

can be represented more generally by replacing the condition **abs(P_{surv} - P_{rpt}) < 10** by the abstract condition **surveySupportsReported(P_{surv}, P_{rpt})**:

wuenic (*C, V, Y, Prpt*) :-
reported(*C, V, Y, Prpt*),
survey(*C, V, Y, Psurv*),
surveySupportsReported(*Psurv, Prpt*) .

The auxiliary predicate used for the abstraction can be defined separately by the rule:

surveySupportsReported(*Psurv, Prpt)* :-
abs(*Psurv - Prpt)* < 10.

The more general rule using the auxiliary predicate **surveySupportsReported** is more flexible than the original rule, because it is compatible with other, and more sophisticated, rules for deciding whether survey data supports government reported data. The use of the more general rule facilitates future refinement of the knowledge base by modifying the auxiliary predicate definitions. For example, the definition of the auxiliary predicate **surveySupportsReported** can be refined to take confidence intervals and other characteristics of the survey into account.

2.5 Negative conditions

In computation logic, as in relational databases, all information is expressed in terms of positive sentences. Facts are expressed by positive atomic sentences, and rules are expressed by conditionals with positive atomic conclusions. Negative information, expressing that something is not the case, is not represented explicitly, but is assumed to hold implicitly if the corresponding positive information cannot be shown. For example, given only the data:

reported(egy, dtp3, 2006, 97).

it is implicit that:

not(reported(egy, dtp3, 2006, 96)).
not(reported(egy, dtp3, 2006, 98)).
etc.

Computational logic, unlike conventional symbolic logic, makes use of this assumption that the negation of an atomic sentence holds if the atomic sentence itself does not hold. This assumption is called the *closed world assumption*.

The closed world assumption makes it possible to derive negative conclusions from facts and rules with positive conclusions. This in turn makes it possible to derive further positive conclusions from rules with negative conditions. For example, if the survey does not support the reported data, the conclusion that the estimate is based on the survey results of 85%

wuenic(egy, dtp3, 2006, 85).

can be derived from the input data:

reported(egy, dtp3, 2006, 97).
survey(egy, dtp3, 2006, 85).

Using the additional rule:

wuenic (*C, V, Y, Psurv*) :-
reported(*C, V, Y, Prpt*),
survey(*C, V, Y, Psurv*),
not(surveySupportsReported(*Psurv, Prpt*)) .

The positive conditions of the rule are satisfied by the input data, and the negative condition is satisfied by the closed world assumption.

2.6 Overriding and refining rules

In some instances it is important to be able to override the current rules when their application gives unacceptable conclusions. For example, it may be desirable to override the default estimate produced by a general rule, by taking account of "shocks to the system" or exceptional events rather than the default estimate produced by the rules that may "dampen" or ignore such events.

In many cases this functionality can be achieved by representing the exceptions themselves by general rules. It can also be achieved more simply, however, by adding working group decisions (wgd) to the knowledge base. These decisions are expressed as atomic sentences using an auxiliary predicate **wgd** having arguments:

wgd(*country, vaccine, year, assigned coverage*)

where ***assigned coverage*** is the working group's estimate, which overrides the coverage that would otherwise be assigned by the rules.

There are many reasons why the working group may decide to override the application of a rule. For example, if a survey does not support the reported data for a given country, year and vaccine, but the same survey does support the reported data for all other vaccines, then the working group could decide that the estimate should be based on the reported results for that vaccine as well (perhaps there was a known problem in calculating coverage for that specific vaccine). Such a working group decision, to assign a reported coverage of 94% to the DTP3 coverage estimate in Egypt in 2007, would be represented as

wgd(egy, dtp3, 2007, 94).

To ensure that the rules are overridden by such exceptional decisions, the rules need to include an extra condition, expressing that there is no overriding working group decision. For example, the rule for the case where survey does not support the reported data has to be revised to:

wuenic (*C, V, Y, Psurv*) :-
not (wgd(*C, V, Y, Pwgd*)),
reported(*C, V, Y, Prpt*),
survey(*C, V, Y, Psurv*),
not (surveySupportsReported(*Prpt, Psurv*)).

An additional rule needs to be added to assign the estimate by means of the working group decision:

wuenic(*C, V, Y, Pwgd*) :-
wgd(*C, V, Y, Pwgd*)

If working group decisions can be generalized, these generalized exceptions can be implemented as rules and included in the knowledge base. The most obvious and direct way to refine a knowledge base is simply to amend a definition of a predicate, replacing it by a more sophisticated definition of the same predicate. However, the representation of knowledge as rules also facilitates refinement by adding rules and by adding conditions to existing rules. The addition of rules for a given predicate extends the rules to cover more cases, whereas the addition of conditions restricts the rules and prevents them from deriving unsatisfactory conclusions. The rules **Accept reported data if there is no reason to exclude it., There is a reason to exclude reported data if it is greater than 100%,** and **There a reason to exclude reported data if the working group decides it should be ignored.** described in section three below illustrate this principle.

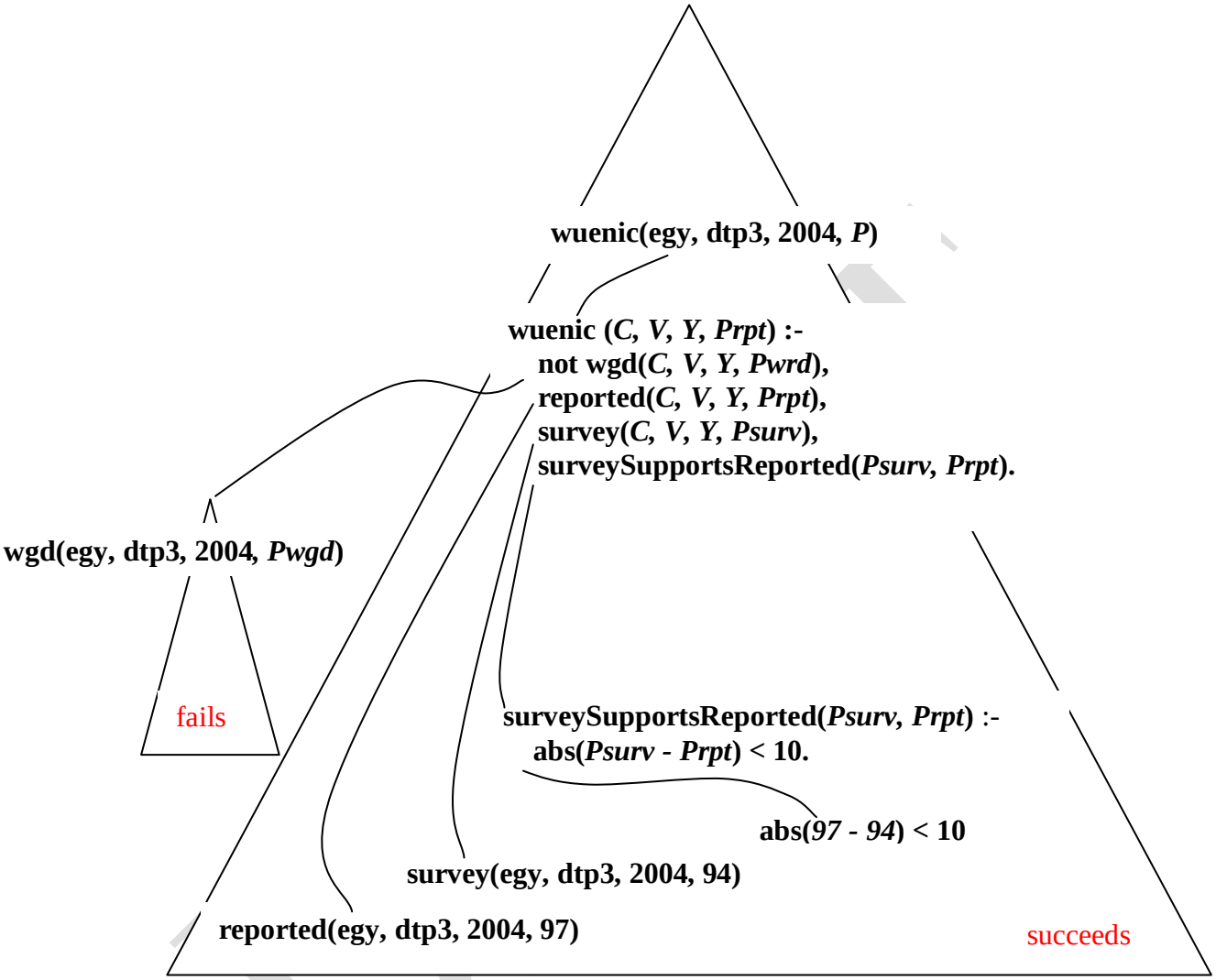
2.7 Reasoning

Much of the power of the computational logic lies in the use of an inference engine which derives logical consequences of information in the knowledge base. These derivations can be viewed in purely logical terms as systematically applying formal rules of logical inference, which are independent of any application domain. In general, the inference engine can be viewed as filling in a triangle, which has the query (or goal) at the apex, atomic data and other information at the base, and domain specific rules in the interior connecting the atoms and the goal.

In general, the inference engine can be viewed as filling in a triangle, which has the query (or goal) at the top, atomic sentences at the bottom and domain specific rules in the interior connecting the atoms and the goal. Some inference engines fill in the triangle top-down; others, bottom-up.

Notice that the top-level goal is to find a value of the variable *P*, such that **wuenic(egy, dtp3, 2004, *P*)** holds for that value.

Figure 1. Inference triangle



For example, given the rules and the data:

```
wuenic (C, V, Y, Prpt) :-
        not(wgd(C, V, Y, Pwgd)),
        reported(C, V, Y, Prpt),
        survey(C, V, Y, Psurv),
        surveySupportsReported(Prpt, Psurv).

wuenic(C, V, Y, Psurv) :-
        not(wgd(C, V, Y, Pwgd)),
```

reported(*C, V, Y, Prpt*),
survey*(C, V, Y, Psurv*),
not(surveySupportsReported(*Prpt, Psurv*)).

wuenic(*C, V, Y, Pwgd*) :-
wgd(*C, V, Y, Pwgd*).

surveySupportsReported(*Prpt, Psurv*) :-
abs(*Psurv - Prpt*) < 10.

reported(egy, dtp3, 2004, 97).
survey(egy, dtp3, 2004, 93.5).

The inference engine derives the value P = 97:

wuenic(egy, dtp3, 2004, 97).

Notice that, in symbolic logic, neither the order in which the rules are written, nor the order in which the conditions of rules are written, affects the results.

2.8 Explanations

The domain-specific rules used to fill in an inference triangle, when made explicit to the user, provide an explanation why the conclusion is a logical consequence of the rules and input data. These explanations are a useful feature which helps to justify the result. If the answer is challenged, then the explanation helps to focus attention on those rules and data that are relevant to the derivation of the answer.

More expressive explanations can be generated as part of the output, by adding an extra argument to the output predicate, **wuenic**. For example,

wuenic(*C, V, Y, P1*, "Reported coverage is supported by survey") :-
not wgd(*C, V, Y, Pwgd,Explanation*),
reported(*C, V, Y, Prpt*),
survey(*C, V, Y, Psurv*),
surveySupportsReported(*Prpt, Psurv*).

The explanation argument is also added to the **wgd** predicate, to justify working group decisions. For example:

wgd(egy, dtp3, 2007, 0.94, "While the reported coverage seems to be supported by survey results, the same survey does not support the reported coverage for other vaccines. The estimate is based on the survey results").

These explanations can be propagated from the working group decisions to the output predicate, using the rule:

wuenic(*C, V, Y, P, Explanation*) :-
wgd(*C, V, Y, P, Explanation*).

For consistency, if an extra argument is added to a predicate in one place, then it must be added to all occurrences of the same predicate. The detailed treatment of explanations in beyond the scope of this paper, and depends in part on the facilities provided by the implementation language. The implementation of explanations in Prolog, for example, is discussed in detail in Bratko[9].

2.9 Further refinement

In our application, the estimation rules are under constant revision and refinement. For example, at the time of writing, the simple rule, which in its earlier incarnation had the form:

wuenic(*C, V, Y, Prpt*) :-
not wgd(*C, V, Y, Pwg*),
reported(*C, V, Y, Prpt*),
survey(*C, V, Y, Psurv*),
surveySupportsReported(*Prpt, Psurv)*.

has now been replaced by the rule:

wuenic(*C, V, Y, Prpt*, "AP:R", "Reported coverage is supported by survey") :-
estimateRequired(*C, V, Y*),
not wgd(*C, V, Y, Pwgd, Action*),
data(reported, *C, V, Y, Prpt*),
survey(*C, V, Y, SurveyDescription, Psurv*),
surveySupportsReported(*Prpt, Psurv)*.

Here the additional arguments of the **wuenic** predicate is the name of the rule used to produce the estimate and the explanation described above. The name of this rule is **AP:R** (for *anchor point*, resolved to *reported* data), for reasons that are explained in the next section. The quotation marks are necessary to override the Prolog convention that expressions starting with uppercase letters are variables.

An additional predicate, **estimateRequired(*C, V, Y*),** is used to specify the country/vaccine/year combinations for which an estimate should be produced. For example, the following facts state that DTP3 estimates should be produced for Egypt for 2004 and 2005 is represented as:

estimateRequired(egy, dtp3, 2004).
estimateRequired(egy, dtp3, 2005).

The **wgd** predicates have been expanded to include a more general ***Action*** argument, which specifies, in addition to the direct assignment of a coverage estimate, other decisions to override the application of the rules. Other decisions include ignoring data for reasons other than those specified in the current rule set, accepting data ignored by the rule set, and adding comments to provide additional explanations.

The **reported** predicate has been replaced by a more general **data** predicate. Other possible values of the first argument are **admin** (for data based on administrative records reported by national authorities) and **gov** (for national authorities' estimate of immunization coverage). These values have proved to be useful for other purposes.

A ***SurveyDescription*** argument has been added to the survey predicate which includes detailed information about the survey, including its title, survey type (e.g., DHS, MICS, EPI cluster survey), year data collected, percent of immunization cards seen, method used to confirm vaccination (e.g., cards, caretaker report, either), age cohort, and sample size. These individual items are extracted from the ***SurveyDescription*** argument using the "in" function.

Some items of the ***SurveyDescription*** argument (e.g., survey title, year of data collection, percent cards seen and sample size) are repeated for each vaccine. A more appropriate representation is to use one predicate having a unique survey identifier and the common items as arguments and a second predicate with the unique survey identifier and vaccine specific details as arguments.

3. Description of the estimation system:

Rules are structured into four levels.

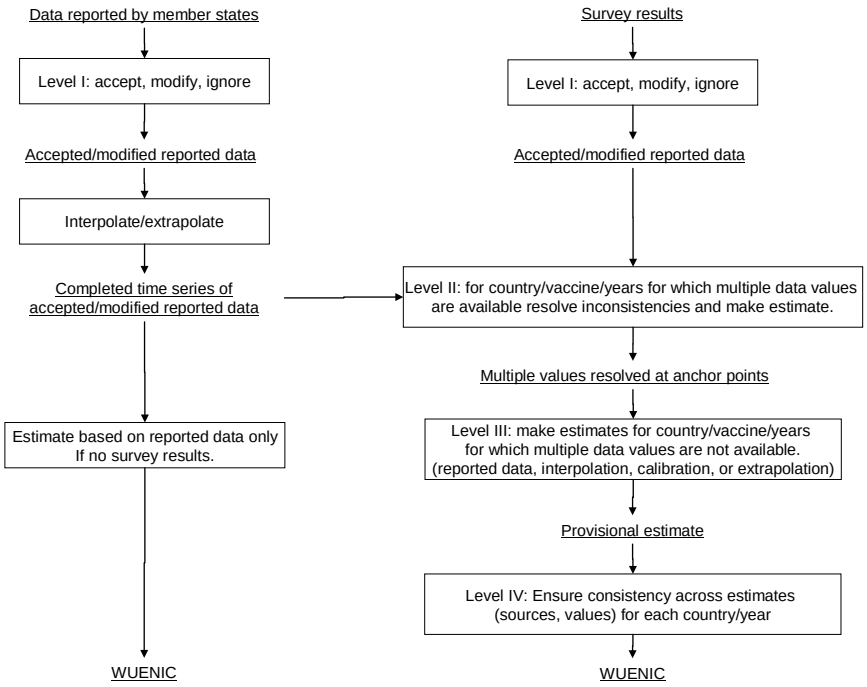
Level one: Accept, modify or ignore reported or survey coverage data.

Level two: Make estimates at “anchor point” years where there is more than one source of data (reported data and survey results). If data are available from only a single source for the entire time series, estimates are made based on these data.

Level three: Make estimates at years between anchor point years and complete the time series.

Level four: Compare estimates for consistency and reconcile discrepancies.

Figure 2: Processing levels.



The general description of the estimation system below is illustrated with examples of the formal description; explanations have been excluded to simplify the examples.

Level one

Each reported and survey data point is passed through a series of "filters" and either a) accepted, b) modified, or c) ignored for further analysis. Two filters for reported data include:

1. Reported coverage figures $\geq 100\%$ are modified or ignored during further analysis. While such reports are theoretically possible they are more likely the result of a calculation error, an inaccurate denominator, or an inaccurate estimate of the number of children immunized (numerator). Reported coverage figures may be modified using an alternative source of denominator data (e.g., United Nations Population Division estimates or more recent census) or replaced by interpolation or extrapolation from reported data less that 100%.
2. While general trends are frequently observed in immunization coverage, it is rare that large changes occur from one year to the next. Such large changes are more likely to be the result of calculation error, missing reports, or the inclusion of children vaccinated during non-routine, supplemental immunization activities. Large jumps in the level of reported data are ignored unless the working group has reasons to believe that the deviation is due to a genuine service delivery change.

Two auxiliary predicates have been created: the first, **reportedAccepted,** represents reported data that is used further in the analysis, and the second,

reportedReasonToExclude, represents specific reasons why a reported data item is excluded. Below are rule samples that exclude reported data greater than 100% and working group decisions to exclude reported data points. The rule to exclude a reported data point if there is a sudden temporal chance is expressed in a similar fashion.

Accept reported data if there is no reason to exclude it.

reportedAccepted(*C, V, Y, Coverage*) :-
data(reported,*C, V, Y, Coverage*),
not(reportedReasonToExclude(*C, V, Y*)).

There is a reason to exclude reported data if it is greater than 100%

reportedReasonToExclude(*C, V, Y*) :-
data(reported,*C, V, Y, Coverage*),
***Coverage* > 100.**

There a reason to exclude reported data if the working group decides it should be ignored.

reportedReasonToExclude(*C, V, Y*) :-
data(reported,*C, V, Y, Coverage*),
wgd(*C, V, Y, Pwgd,*, ignoreReported).

To facilitate comparison of reported data with survey results and other information, a complete time series is constructed based solely on reported data for all country/vaccine/year combinations for which estimates are required. Its values are the accepted reported data that exist for any given year. If there are years between two accepted points for which there is no accepted value, the time series value for that year is estimated using linear interpolation.

The value for the reported time series is the reported data if there is reported data and it has been accepted.

reportedTimeSeries(*C, V, Y, Coverage*) :-
reportedAccepted(*C, V, Y, Coverage*).

The reported time series value in a year for which there is no accepted reported data is derived by interpolating between the accepted reported values for years before and after the year for which there is no accepted reported data.

reportedTimeSeries(*C, V, Y, Coverage*) :-
not(reportedAccepted(*C, V, Y,P*)),
reportedAccepted(*C, V, Ybefore, Pbefore*),
reportedAccepted(*C, V, Yafter, Pafter*),

Y > Ybefore ,
Y < Yafter,
not(reportedAcceptedBetween(*C, V, Ybefore, Yafter*)),
interpolate(*Ybefore, Pbefore, Yafter, Pafter, Y, Coverage*).

Because multiple accepted reported data points preceding and following the year with missing data are possible, the auxiliary predicate **reportedAcceptedBetween** is used to determine if there is at least one reported data point that has been accepted between two years. The negation of this predicate identifies the years closest, before and after, to the year for which accepted data are missing.

reportedAcceptedBetween(*C, V, EarlyYear, LateYear)* :-
reportedAccepted(*C ,V, Y, P*),
Y > EarlyYear,
Y < LateYear.

If the other conditions of the rule apply, the rule to interpolate between the identified years is applied.

interpolate(*Yearbefore, Pbefore, Yearafter, Pafter, Yearinter, Pinter*) :-
Pinter* is *Pbefore + (Yearinter - Yearbefore) * ((Pafter - Pbefore) / (Yearafter - Yearbefore)).

Extrapolation is used to estimate missing data from the earliest reported data back to the beginning of the time series and from the latest reported data forward to the end of the time series.

Survey results are also accepted, modified, or ignored during further analysis. Surveys with sample sizes < 300 or outside the appropriate age cohort are ignored unless the working group has other reasons to accept the results. If adequate data are available, results for multi-dose antigens (e.g., DTP3, Pol3, etc) are modified for recall bias. For example, the rule to exclude a survey because the sample size is less than 300 is written as:

Accept survey data if there is no reason to exclude it.

surveyAccepted*(C, V, Y, Coverage) :-*
survey*(C, V, Y, SurveyDescription, Coverage),*
not(surveyReasonToExclude*(C, V, Y, Coverage)).*

There is a reason to exclude a survey if the sample size is less than 300 and the working group has not decided to accept the survey.

surveyReasonToExclude(*C, V, Y, Coverage*) :-

survey(*C, V, Y, SurveyDescription, Coverage*),
***SampleSize* in *SurveyDescription*,**
***SampleSize* < 300,**
not(wgd(*C, V, Y, Pwgd,* acceptSurvey)).

Additional conditions (e.g., inappropriate age cohort, working group decision if survey results are compromised by design or implementation issues) may also lead to surveys being excluded. As with reported data, there is an auxiliary predicate, **surveyAccepted** which evaluates whether survey results should be used in further analysis. The rules allow working group decisions to override rules that would exclude data points for both reported and survey data. For example, while a survey with a sample size of 299 would be ignored by one of the processing rules, the working group can decide to accept the results and reinstate the survey.

Level two

In cases where the only source of data for a country are reports from national authorities, and the working group has no reason to ignore these reports, estimates are based on completed time series of accepted reported data.

For a given country and vaccine, if both survey results and data reported by the national authorities are available, estimates are first made at "anchor point" years where there are multiple sources of data. At these points survey results may support reported data or they may be significantly different. If reported data are within 10% points of survey results and there is no working group decision invoking other considerations, the estimate is based on reported data for that year. In computational logic the rule is written as:

The value at a year where there are survey results is the accepted reported data if there is no working group decision assigning an anchor point value, and the survey results supports the value of the time series of reported accepted data.

anchorPoint(*C, V, Y, Preported*) *:*
not(wgd(*C, V, Y, Pwgd, assignAnchor*)),
reportedTimeSeries(*C, V, Y, Preported),*
surveyAccepted(*C, V, Y, Psurvey*),
surveySupportsReported(*Preported, Psurvey*).

The auxiliary predicate **surveySupportsReported** is represented as described in section two,above.

The rule

The value at a year where there is survey data is the accepted survey results if there is no working group decision assigning an anchor point value, and the survey results does not support the value of the time series of reported accepted data.

anchorPoint(*C, V, Y, Psurvey*) :-
not(wgd(*C, V, Y, Pwgd,* assignAnchor)),
reportedTimeSeries(*C, V, Y, Preported*),
surveyAccepted(*C, V, Y, Psurvey*),
not(surveySupportsReported(*Preported, Psurvey*)).

sets the value in the anchor point year equal to the survey results. The rule

The value at year is set by the working group.

anchorPoint(*C, V, Y, Pwgd*) :-
wgd(*C, V, Y, Pwgd,* assignAnchor).

allows the working group to assign a value to an anchor point year.

A coverage estimate at anchor points is assigned by the rule:

The coverage estimate is the value established at the anchor point years if an estimate is required and there is no working group decision assigning a coverage estimate.

wuenic(*C, V, Y, Coverage*) :-
estimate_required(*C, V, Y*),
not(wgd(*C, V, Y, Pwgd,* assignWUENIC)),
anchorPoint(*C, V, Y, Coverage*).

Level three

Estimates for years between two anchor point years depend on the way in which estimates are resolved at the anchor points. If surveys support reported data at both anchor point years, then the estimates between the anchor point years are the reported data as accepted or modified in level one. Otherwise, the estimates are the accepted or modified reported data calibrated to the level of the estimates at the anchor point years. Alternatively the working group may decide to interpolate between the anchor point estimates providing an accompanying justification.

For example an estimate between two anchor points, at least one of which has not been resolved to the reported time series value is the reported time series value calibrated to the level of the surveys.

The estimate is the reported time series value calibrated to the level of the surrounding anchor point values if an estimate is required, there is no working group decision assigning an estimate, there are anchor points surrounding the

estimate year, and at least one of the anchor point values has been resolved to a value different from the reported time series value.

wuenic(*C, V, Y,* ReportedCalibrated) :-
estimateRequired(*C, V, Y*),
not(wgd(*C, V, Y, Pwgd*, assignWUENIC)),
anchorPoint(*C, V, Ybefore, PanchorBefore*),
anchorPoint(*C, V, Yafter, PanchorAfter*),
***Y* > *Ybefore* ,**
***Y* < *Yafter*,**
not(anchorPointBetween(*C, V, YBefore, YAfter*)),
not(bothAnchorsReported(*C, V, YearBefore, PanchorBefore, YearAfter, PanchorAfter*),
calibrateBetween(*C, V, Ybefore, Yafter, Y, ReportedCalibrated*).

Both anchor point values equal to reported time series values.

bothAnchorsReported(*C, V, YearBefore, PanchorBefore, YearAfter, PanchorAfter*) :-
reportedTimeSeries(*C, V, YearBefore, PanchorBefore*),
reportedTimeSeries(*C, V, YearAfter, PanchorAfter*).

Reported time series value between to anchor points calibrated to the level of the anchor point values.

calibrateBetween(*C, V, YearBefore, YearAfter, Year, ReportedCalibrated*) :-
reportedTimeSeries(*C, V, YearBefore ,PRBefore*),
reportedTimeSeries(*C, V, ,YearAfter, PRAfter*),
anchorPoint(*C, V, YearBefore, AnchorBefore*),
anchorPoint(*C, V, YearAfter, AnchorAfter*),
reportedTimeSeries(*C, V, Year, ReportedTimeSeriesValue*),
interpolate(*YearBefore, PRBefore, YearAfter, PRAfter, Year, ReportedInterpolated*),
interpolate(*YearBefore, AnchorBefore, YearAfter, AnchorAfter,Year, AnchorInterpolated*),
***Adj* is *AnchorInterpolated - ReportedInterpolated*,**
***ReportedCalibrated* is *ReportedTimeSeriesValue + Adj*.**

Level four

Levels 1 through 3 operate on data for each country and vaccine independently. Estimates across vaccines are reconciled in Level 4. For example, in some countries DTP1 coverage is underreported because it is not considered the "final" of the three dose DTP series recommended in many national schedules. If DTP3 coverage levels are greater than DTP1 levels or no DTP1 results have been reported, DTP1 is estimated based on a second degree polynomial function describing the relationship between DTP3 coverage

and the difference between DTP1 and DTP3 (DTP1 coverage - DTP3 coverage). This function and the values for the coefficients were estimated based on a review of 282 surveys from 101 countries published between 1980 and 2004.

DTP1 coverage estimates are based on relationship between DTP1 and DTP3 coverage observed in 282 surveys conducted in 101 countries between 1980 and 1999 if an estimate is required and there is no working group decision to assign DTP1 coverage and DTP3 coverage is greater than DTP1 coverage.

```
wuenic(C, dtp1, Y, DTP1) :-
        estimateRequired(C, dtp1, Y),
        not(wgd(C,dtp1, Y, Pwgd, assignWUENIC)),
        wuenic(C, dtp1, Y, Dtp1Cov),
        wuenic(C, dtp3, Y, Dtp3Cov),
        Dtp3Cov > Dtp1Cov,
        DTP1 is Dtp3Cov + (-0.0066 * (Dtp3Cov * Dtp3Cov)) +
                    (0.4799 * Dtp3Cov) + 16.6,
        correctForDTP399(DTP1, Dtp3Cov).
```

Implementation

The formal description above has been implemented for automated production. Data and information (administrative data, estimates made by national authorities, survey results and working group decisions) are maintained in a Microsoft Access[10] production database. Rules are written in SWI Prolog[11]. An R[12] programme extracts data from the Access data base and creates a country-specific file of Prolog predicates of the data, information and working group decisions. SWI Prolog executes the rules using the country-specific file of data and information and produces a file of estimates with the supporting data and working group decisions. An R programme reads this file and outputs graphs and LaTex[13] source code of a country-specific summary. LaTex is used to produce country-specific Portable Document Format[14] (PDF) formatted reports. Once data and working group decisions have been updated it takes approximately 20 seconds to produce each country-specific report. Alternative implementations are certainly possible.

4. Discussion

Rule-based[15,16,17] and expert systems[18] based on the logical evaluation of facts and rules have been described for forecasting and there are passing reference to such systems in global health[19] and demography[20]. We are not aware, however, of any applications that provide representation and decision support for the production of quantitative estimates that rely on a variety of multiple incomplete data sources.

Our estimation methods remain judgmental and incorporate context-specific, local knowledge about singular events. The process of formalization described above allows us to produce consistent and replicable estimates using transparent methods while

continuing to draw on data from a variety of sources and to incorporate context-specific information. Formalization facilitates documentation of estimates based on context-specific information as well as generally applicable rules.

Formalization requires that vague notions such as "supports" or "is consistent with" be operationalized. We have addressed such concepts by providing a precise operational rule for such concepts. An alternative approach is to describe such concepts using fuzzy set theory.[21] Haack[22] provides a critique of these two approaches.

It is necessary to ensure that the interaction between rules provides consistent results. For example, the rule set should not general multiple estimates for a given country/vaccine/year combination and should generate an estimate for each required combination. The development of a comprehensive test suite is essential during development of such a formal system.

It is important to note that the formalized, explicit rules are intended to assist the working group in making consistent, replicable, transparent and documented estimates. The intent is not to delegate the estimation process to a mechanical procedure. Rules are applied to particular cases. If a conclusion is unacceptable or if there is disagreement regarding the conclusion, arguments that the rule should not apply in this case are sought. If persuasive arguments are found, then an exception may be made for the specific case or the general rule may be revised.

The use of computational logic has an advantage over normal relational database systems in that we can include both simple facts (e.g., 2004 reported DTP3 coverage Egypt of 97%) as well as rules that allow us to generalize or infer information from the simpler facts in our knowledge base. Exceptions may be stated either as facts or as rules in their own right.

Formalizations in computational logic can be fairly easily implemented in a variety of programming languages. As seen above predicates resemble data structures and rules are expressed in logic that is easily represented in both Prolog and expert system shells. We believe that it would be interesting to implement our formalization in the database query and programming language SQL[23].

The use of computational logic extends the flexibility of standard statistical methods by providing tools to incorporate exceptions and expert judgement in a transparent and replicable fashion. We believe that formalization using computational logic can be usefully applied to a wide range of official statistics.

Appendix: Knowledge representation and reasoning using computational logic: an annotated bibliography

Kowalski[4] presents a comprehensive, informal introduction to computational logical for a general audience. Brachman - Levesque[24] provide an introduction to knowledge representation and reasoning. Russell - Norvig[25] is a popular university-level text on artificial intelligence and chapters 7 through 9 present material on knowledge representation and reasoning. Sowa[26] provides a readable overview of the field. Davis[27] contains practical advice on representing domain knowledge in formal logic.

Computational logic is an extension of first order predicate logic. Logic has long been used to unambiguously represent knowledge, and computers can efficiently prove theorems expressed in the clausal form of logic. A readable introduction to logic is Bennet[28]. Barwise - Etchemendy[29] is a classroom text with a computer science approach and includes a CD-ROM of interactive exercises. Enterton[30] provides a more advanced approach to logic. Genesereth and Nilisson[31] is a classic text on the application and foundations of logic in artificial intelligence. Kowalski[8] presents the use of clausal logic for representing knowledge and general problem solving.

---

[1] United Nations Department of Social and Economic Affairs/Population Division (2006). Methodology of the United Nations population estimates and projections. In: *World Population Prospects: The 2004 Revision, Volume III: Analytical Report.* New York: United Nations.

[2] UNICEF, WHO, The World Bank, and the UN Population Division (2007). *Levels and Trends of Child Mortality in 2006: Estimates developed by the Inter-agency Group for Child Mortality Estimation.* New York.

[3] United Nations Development Programme. Reader's Guide (2009). In Human Development Report: 2009. Overcoming barriers: Human Mobility and development. New York: United Nations Development Programme.

[4] Kowalski R. (in press). Computational Logic and Human Thinking: How to be Artificially Intelligent. Cambridge University Press. Draft available at: http://www.doc.ic.ac.uk/~rak/papers/newbook.pdf.

[5] Robinson J.A. (2000). Computational Logic: Memories of the Past and Challenges for the Future. In J. Lloyd, V. Dahl, U. Furbach, M. Kerber, K.-K. Lau, C. Palamidessi, L. Moniz Pereira, Y. Sagiv, and P.J. Stuckey, editors, *Proceedings First International Conference on Computational Logic (CL2000)*, volume 1861 of *Lecture Notes in Artificial Intelligence*, Springer.

[6] Burton A., Monash R., Lautenbach B., Gacic-Dobo M., Neill M., Karimov R., Wolfson L., Jones G., Birmingham M. (2009). WHO and UNICEF estimates of national infant immunization coverage: methods and processes. Bulletin of the World Health Organization., 87,535–541.

[7] Colmerauer, A. and Roussel, P. (1992). The birth of Prolog. *The second ACM SIGPLAN conference on History of programming languages*, p. 37-52.

[8] Kowalski R. (1979). Logic for Problem Solving. North-Holland. Available at: http://www.doc.ic.ac.uk/~rak/.

[9] Bratko I. (2000). Prolog Programming for Artificial Intelligence (3d edition). Addison Wesley.

[10] Microsoft Access reference.

[11] Wielemaker J., Schrijvers T., Triska M., Lager M. (2010). SWI-Prolog. Submitted, Theory and Practice of Logic Programming.

[12] R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org.

[13] Lamport, Leslie (1994). *LaTeX: A document preparation system: User's guide and reference*. illustrations by Duane Bibby (2nd ed.). Reading, Mass: Addison-Wesley Professional.

[14] Adobe Systems Incorporated (2006), PDF Reference, Sixth edition version 1.23.

[15] Armstrong J., Adya M., and Collopy F. (2001). Rule-based forecasting: using judgement in time-series extrapolation in Armstrong J. Principles of forecasting. Kluwer Academic Publishers.
[16] Collopy F. and Armstrong J. (1992). Rule-based forecasting: development and validation of an expert systems approach to combining time series extrapolations. Management Science, 38:1394-1414.
[17] Rule-Based Forecasting (1990). Retrieved 2 December 2010 from http://userpages.umbc.edu/~adya/rbf.html.
[18] Collopy F., Adya M., and Armstrong J. (2001). Expert systems for forecasting in Armstrong J. Principles of forecasting. Kluwer Academic Publishers.
[19] Sekhri N (2007). Forecasting for Global Health; New Money, New Productions & New Markets. Retrieved 23 September 2009, from http://www.cgdev.org/doc/ghprn/Forecasting_Background.pdf
[20] Bijak J (2006), "Forecasting International Migration: Selected Theories, Models and Methods," Retrieved 23 September 2009, from http://www.cefmr.pan.pl/docs/cefmr_wp_2006-04.pdf.
[21] Zadeh L (1965). *Fuzzy sets. Information and Control*. 1965; 8: 338–353.
[22] Haack S. (1974). Deviant Logic. University of Chicago Press.
[23] Chamberlin, Donald D; Boyce, Raymond F (1974). SEQUEL: a structured English query language in *Proceedings of the 1974 ACM SIGFIDET Workshop on Data Description, Access and Control* (Association for Computing Machinery): 249–64.
[24] Brachman R. and Levesque H. (2004). Knowledge Representation and Reasoning. Morgan Kaufman.
[25] Russell S. and Norvig P. (2009). Artificial Intelligence: A Modern Approach. Prentice Hall.
[26] Sowa J. (2000). Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole..
[27] Davis E. (1999). Guide to axiomatizing domains in first-order logic. Electronic Newsletter on Reasoning about Action and Change, 99002, 1999. Retrieved 2 December 2010 from http://cs.nyu.edu/davise/guide.html.
[28] Bennett D. (2004). Logic made easy. W. W. Norton & Company.
[29] Barwise J and Etchemendy J. (2002). Language, Proof and Logic. Center for the Study of Language and Information.
[30] Enderton H.(2001). A Mathematical Approach to Logic (2d edition). Academic Press.
[31] Genesereth M. and Nilisson N. (1987). Logical Foundations of Artificial Intelligence. Morgan Kaufmann Publishers.