medexter
clinical decision support

Arden Syntax and fuzzy medical linguistic concepts in Fuzzy Arden Syntax

An introduction

Jeroen S. DE BRUIN,
Andrea RAPPELSBERGER, and
Klaus-Peter ADLASSNIG

Medexter Healthcare
Borschkegasse 7/5
A-1090 Vienna

jb@medexter.com

eHealth Summit Austria — 23–24 May 2017

Arden Syntax

- Knowledge representation standard capable of the computerized representation of medical knowledge
- Medical rule encoding in a syntax resembling natural language
  - Code better understandable
  - Code easier verifiable
- Units of knowledge are called medical logic modules (MLMs)
- Specification freely available under http://www.hl7.org/implement/standards/product_brief.cfm?product_id=372

```
maintenance:
    title:          Positive Microbiology Result;;
    mlmname:        PositiveMibi;;
    arden:          version 2.9;;
    version:        1.0;;
    institution:    Medexter Healthcare, Vienna, Austria;;
    author:         knowledge engineering group;;
    specialist:     clinical specialists group;;
    date:           2017-01-11;;
    validation:     production;;
library:
    purpose:        ;;
    explanation:    ;;
    keywords:       ;;
    citations:      ;;
    links:          ;;
knowledge:
    type: data_driven;;
    data:
        (sampleID, testID, pathogenID, pRes) := Argument;
        ;;
    priority:       ;;
    evoke:          ;;
    logic:
        // Qualitative result
        cRes := localized 'positive';
        if pRes is cRes then
            conclude true;
        endif;
    ;;
    action:
        return true;
    ;;
    urgency: ;;
resources:
    default: de;;
    language: en
       'positive': "positive";
    ;;
```

History

Version	Year	Important changes
2.1	2002	new string operators; reserved word "currenttime" returns the system time
2.5	2005	object capabilities: create and edit objects; XML representation of MLMs (except logic, action and data slot)
2.6	2007	UNICODE encoding; additional resources category to define text resources for specific languages; time-of-day and day-of-week data types; "localized" operator to access texts in specific languages
2.7	2008	enhanced assignment statement; extended "new" operator to allow easy and flexible object instantiation
2.8	2012	additional operators for list manipulation; operators to manipulate parts of given date and time values; switch statements; keyword "breakloop" for aborting a loop; number of editorial corrections
2.9	2013	**Fuzzy**: fuzzy data types, fuzzy sets, and fuzzy logic; adjustment of all available operators to be able to handle fuzzy data types
2.10	2014	XML representation of MLMs (including logic, action and data slot)

Fuzzy logic

Traditional logic is bivalent, which means that only two truth values are allowed: every proposition must be either true or false. But the inherent vagueness of many terms, apparent in the sorites paradox, suggests that this requirement is too rigid if logic is to encompass the full scope and complexity of natural language.

Fuzzy logic has been developed, initially by the computer scientist Lofti Zadeh, to allow for imprecision and degrees of truth. Truth is presented as a continuum between true (1) and false (0). So, for instance, a particular proposition that is 'partly true' or 'more or less true' might be represented as true to degree 0.8 and false to degree 0.2. Fuzzy logic has been particularly important in AI (artificial intelligence) research, where 'intelligent' control systems need to be responsive to the imprecisions and nuances of natural language.

From: Dupré, B. The sorites paradox. In Dupré B. (2007) *50 philosophy ideas you really need to know*. Quercus Publishing PLc, London, p. 122.

Fuzzy Arden Syntax: Modelling uncertainty in medicine

- **linguistic uncertainty**
  - due to the unsharpness (fuzziness) of boundaries in linguistic concepts; gradual transition from one concept to another
  - modeled by fuzzy sets (e.g., fever, increased glucose level, hypoxemia)

- **propositional uncertainty**
  - due to the incompleteness of medical conclusions; uncertainty in definitional, causal, statistical, and heuristic relationships
  - **here**: modeled by truth values between zero and one (e.g., 0.6, 0.9)

Medical concepts and guidelines

- Medical concepts are expressed in natural language
  - Often uncertain with respect to its meaning / range / boundaries
  - Subjective to individual interpretation, especially with comparative or subjective criteria

- Use case: ECDC definition of bloodstream infection

3.2 Bloodstream infection

- Patient has at least one positive blood culture for a recognised pathogen
  – or –
- Patient has at least one of the following signs or symptoms: fever (> 38 °C), chills, or hypotension
  and
  two positive blood cultures for a common skin contaminant (from two separate blood samples, usually within 48 hours).

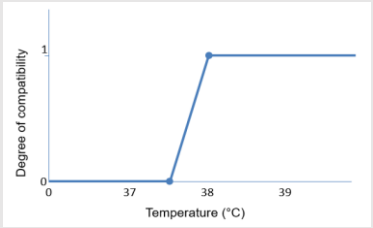
- Chills, ICD-10
  - A sensation of cold that often marks the start of an infection and the development of a fever.
  - The sudden sensation of being cold. It may be accompanied by shivering.

Fuzzy sets

- Formalism to model classes with linguistic uncertainty, e.g., medical concepts
- Mathematical sets, whose elements have a ***truth value***, expressed as a number between (and including) 0 and 1 (c.f., "normal", or "crisp" sets, have a truth value of *either* 0 *or* 1)
- Each set has a membership function that determines the truth value of an element in the problem space with that set.

Linguistic medical concept

Membership function



Fuzzy set

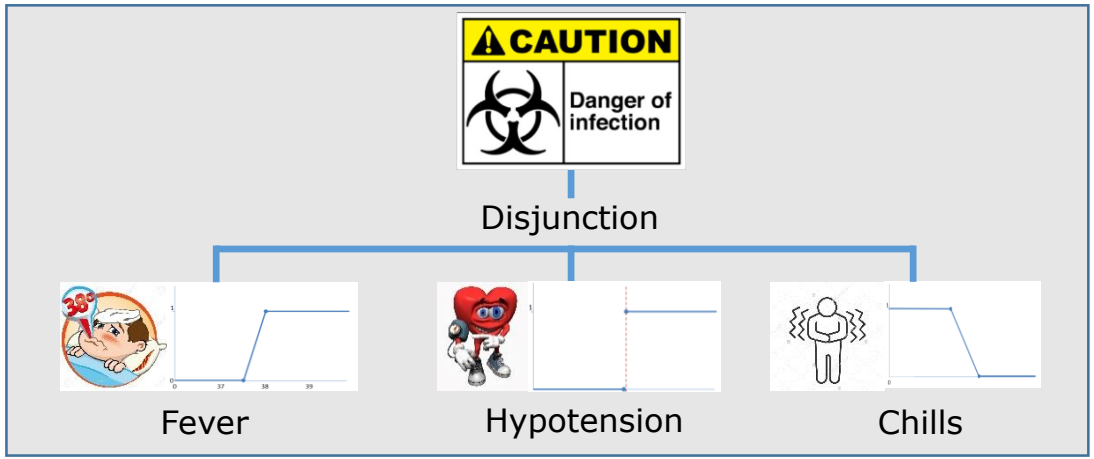
Patient	Temp.	Comp.
Katharina	37.3 °C	0
Julia	37.8 °C	0.6
Alexander	38.2 °C	1

Fuzzy logic

- Algebra used to evaluate logical combinations of concepts that are assigned a truth value, possibly with a truth value as result

- Redefinition / extension of well-known logical operators
  - Negation
  - Conjunction
  - Disjunction
  - …

- **Example:** ECDC clinical signs of blood stream infection



Arden Syntax vs. Java …

```
knowledge:
    type: data_driven;;
    data:
        testID:= 123;
        (temperature,heartRate,respRate,PaCO2,WBcellCount,immatureBand)
        := READ {SELECT temperature, heartRate, respRate, PaCO2, WBcellCount, immatureBand
            FROM sirsvalues WHERE IDPatient = testID};
        ;;
    priority: ;;
    evoke: ;;
    logic:
        //Start - Checking SIRS criteria
        counter:=0;

        if temperature is greater than 38 or temperature is less than 36 then
            counter:= counter + 1;
        endif;

        if heartRate is greater than 90 then
            counter:= counter + 1;
        endif;

        if respRate is greater than 20 or PaCO2 is less than 32 then
            counter:= counter + 1;
        endif;

        if WBcellCount  is greater than 12000 or WBcellCount is less than 4000
        or immatureBand is greater than 10 then
            counter:= counter + 1;
        endif;

        if counter is greater than or equal 2 then
            notification:= localized 'SIRS';
            conclude true;
        endif;
        //End - Checking SIRS criteria
```

```
import java.util.Locale;
import java.util.ResourceBundle;
import java.sql.*;
import java.text.MessageFormat;

public class sirs_notification {
    // JDBC driver name and database URL
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/SIRSDB";

    //  Database credentials
    static final String USER = "root";
    static final String PASS = "R00tPassW0rd";

    // Database Connection
    static Connection conn = null;

    private static boolean setupConnection()
    {
        boolean retval = true;

        try{
            // Register JDBC driver
            Class.forName(JDBC_DRIVER);

            // Open a connection
            conn = DriverManager.getConnection(DB_URL, USER, PASS);
        } catch(Exception e){
            //Handle errors for Class.forName
            e.printStackTrace();
             retval = false;
        }

        return retval;
    }

    public static void main(String... args) throws Exception
    {
        final String sqlQuery = "SELECT temperature, heartRate, respRate, PaCO2, WBcellCount, immatureBand "
                + "FROM sirsvalues WHERE IDPatient = {0}";

        if (!setupConnection())
            return;

        // Execute a query
        Statement stmt = conn.createStatement();
        String sql = MessageFormat.format(sqlQuery, args[0]);
```

… and more Java.

```
43              return;
44
45          // Execute a query
46          Statement stmt = conn.createStatement();
47          String sql = MessageFormat.format(sqlQuery, args[0]);
48
49          ResultSet rs = stmt.executeQuery(sql);
50
51          // Extract data from result set
52          while(rs.next()) {
53              //Retrieve by column name
54              int temperature = rs.getInt("temperature");
55              int heartRate = rs.getInt("heartRate");
56              int respRate = rs.getInt("respRate");
57              int PaCO2 = rs.getInt("PaCO2");
58              int WBcellCount = rs.getInt("WBcellCount");
59              int immatureBand = rs.getInt("immatureBand");
60
61              // Start - Checking SIRS criteria
62              int counter = 0;
63
64              if (temperature > 38 || temperature < 36)
65                  counter++;
66
67              if (heartRate > 90)
68                  counter++;
69
70              if (respRate > 20 || PaCO2 < 32)
71                  counter++;
72
73              if ((WBcellCount > 12000 || WBcellCount < 4000) || immatureBand > 10)
74                  counter++;
75
76              // End - Checking SIRS criteria
77
78              // Refer to a .properties file, one for each language
79              Locale currentLocale = new Locale(args[1], args[2]);
80              ResourceBundle messages = ResourceBundle.getBundle("MessagesBundle", currentLocale);
81
82              if (counter >= 2)
83                  System.out.println(messages.getString("sirs_detected"));
84              else
85                  System.out.println(messages.getString("no_sirs_detected"));
86          }// while
87      }
88  }
89
90
```

Fuzzy Arden Syntax

- Fuzzy constructs inherently supported by Arden Syntax version 2.9 and later

- Changes that facilitate fuzziness, e.g., for fuzzy sets
  - Incorporation of `truth value` data type and extension of the crisp truth value model to a fuzzy model
  - Incorporation of the `fuzzy set` data type
  - Extended syntax for automatic truth value calculation based on fuzzy sets (`is in`)

```
knowledge:
    type: data_driven;;
    data:
        patientID := Argument;

        if patientID is string then
          patientID := "'" || patientID || "'";
        endif;

        (hba1c, fpg, rpg, ogtt)
        := READ {SELECT SCREENING_DM_HBA1C, SCREENING_DM_FPG, SCREENING_DM_RPG, SCREENING_DM_OGTT2H
            FROM Screening WHERE IDPatient = patientID};

        // Fuzzy set definitions
        fs_ifg_ogtt := fuzzy set (7.7, 1), (7.9 ,0);
        fs_ifg_fpg := fuzzy set (5.9, 0), (6.1, 1), (6.9, 1), (7.1, 0);

        fs_iigt_fpg := fuzzy set (6.0, 1), (6.2 ,0);
        fs_igt_fpg := fuzzy set (7.0, 1), (7.2 ,0);
        fs_igt_ogtt := fuzzy set (7.6, 0), (7.8, 1), (11.0, 1), (11.2, 0);
        ;;
    priority: ;;
    evoke: ;;
    logic:
        // Start - Checking for impaired glucose regulation
        // Impaired fasting glycaemia
        if fpg is in fs_ifg_fpg and ogtt is in fs_ifg_ogtt then
            notification := localized 'IFG';
            conclude true;
        endif;
        conclude true;
        //End - Checking for impaired glucose regulation
        ;;
    action:
        return notification;
        ;;
```