CS 514 Project 2

Domain:

The domain I choose is an automated system for investment advice.
This type of expert system helps people to manage their assets. The aim is to maximize the expected return and minimise the risk when determining asset allocation.
However, a portfolio is generally designed according to the investor's risk tolerance, time frame and investment objectives. The monetary value of each asset may influence the risk/reward ratio of the portfolio.

This type of expert system is widely used for tasks like assets portfolio management or stock portfolio management. Similarly, it can be used for finding best combination bets. The reasoning task I want to automate is to find the best combination that satisfies the gambler's risk tolerance from given bets.

Fuzzy Logic is particularly useful in this kind of system since we don't know the chance of winning for most of the time. We can't give a crisp winning probability, we can only know some teams are more likely to win since they performed better. And this is exactly where fuzzy logic does the best.

Task:

A simple sport betting advisor
It's automated reasoning works as follows

1. Read preference provided by the user
2. Analyze the bets by calculating their fuzzy winning chance based on the team's recent performance. Then it will calculate the risk according to the investor's bet amount and the winning chance.
3. Print out the result.

Example:

I created three bets (in example.clp) and their risks will be calculated based on the given preference (in example.clp) and the rating of teams (in facts.clp)

Variables used in the program:

Templates

Common
match-score: Slot for computed fuzzy-match score. It will be used to determine which result fits better.

bet-type: The type of bet provided by the provider. For example, a straight wager is a bet on the outright winner of a game or event.

bet-type-arg: The argument required by different types of bets. For example, a straight wager has two options Win or Lose.

provider: The betting company that provides the service.

Preference
pid: id of this preference

sport: The prefered sport type

currency: The prefered currency.

bet-amount: The prefered bet amount.

Bet
bid: id of this bet

odds: The odds of a bet, it should be larger than 1. Here we use Decimal or Continental odds, the ratio of total paid out to stake: Ex 6.0, 2.0, 1.5

winning-rate: The winning rate of a bet. It will be computed during the process.

timeslot-start/end: Starting/ending time (in epoch time) of a bet.

currency: The currency provided by the provider.

BetAnalysis

winningChanceFv: Slot for winning chance fuzzy value.

Team
name: Name of this team.

rate: Rate of a team based on their recent performance.

sport: Sport played by this team.

TeamAnalysis
rate: same as Team template.

rateFv: Slot for the rate of team fuzzy value.

ExpectedFuzzy
expectedFv: Slot for the expected value.

RiskFuzzy
riskFv: Slot for the risk fuzzy value.

Fuzzy Variables

?*riskFvar* : Fuzzy variables for the risk. From 0 to 10, 0 is no risk and 10 means very risky.

?*betAmountFvar* : Fuzzy variables for bet amount. From 0 to 10000.0 usd, 0 is very low and 10000 means very high.

?*winningChanceFvar* : Fuzzy variables for winning chance. From 0 to 1, 0 is very low and 1 means very high.

?*expectedFvar* : Fuzzy variables for expected value (odds times winning chance). From 0 to 2, 0 is very low and 2 means very high.

?*teamRateFvar* : Fuzzy variables for the rate of team. From 0 to 10, 0 is very bad and 10 means very good.

Usage:

1. Set classpath to where your FuzzyJ located. In .classspath file: <classpathentry kind="lib" path="<path to fucczy>/fuzzyJ-2.0.jar"/>.
2. Set Jess main class to “nrc.fuzzy.jess.FuzzyMain”
3. Unzip, run (batch “<path-to-directory>/exampleX.clp”) in Jess shell
4. Have fun!!