6.871 Spring 2005
Final Project Paper 5/12/2005

# How to Score Free Drinks From Guys

## 1. Motivation and Task

We were motivated by the people's behaviors and social interactions at bars. Specifically, we noticed that certain women always have free drinks lined up, certain women always end up meeting and conversing with new people, certain men always comment on how short someone's skirt is, different men respond differently to various flirting styles, and that behaviors and dress can vary depending on the location. Intrigued by these patterned observations and reasoning behind human interactions at bars, we decided to capture that knowledge in our final project.

We propose to build an advisor expert system that will help women get free drinks at bars. When the user, a woman, walks into a night scene, she will target a prospective man from whom she wishes to receive a drink. She will use our system by inquiring it for recommended course of actions. The system will, in response, pose a series of customized questions to the woman to collect information about the woman's physical attributes and profession, the prospective man's behaviors and state, and the venue. Based on the inputs, the system will make several levels of inferences to make some conclusions about the woman's attractiveness, personality traits, the man's traits, and the setting characteristics. Using these intermediate inferences, the system will eventually result in a recommend course of action for the woman in her specific scenario.

An example of user-system dialogue might look like this:

```
(ask [action ann ?x] #'suggest)
What is ANN's age: 40
What is ANN's height: 61
What is ANN's weight: 150
Is it the case that ANN wears glasses: Yes
Is it the case that ANN has braces: No
Is it the case that ANN has blue eyes: Yes
Is it the case that ANN has blonde hair: Yes
What is ANN's hair length: Long
What is ANN's shoe type: Heels
What is ANN's clothing fit: Loose
Is it the case that ANN is wearing a skirt: No
What is ANN's profession: Education
What is ANN's location(venue type): Club
Is it the case that ANN's target man is wearing a suit: Yes
Is it the case that ANN's target man is alone: Yes
Is it the case that ANN's target man is speaking loudly: Yes
Is it the case that ANN's target man is making large gestures: Yes
[ACTION ANN ASK-THE-BARTENDER-FOR-A-MENU-AND-KNOCK-OVER-YOUR-DRINK-
WHEN-YOU-REACH-FOR-IT] 1.0
```

```
[ACTION ANN PLACE-YOUR-GLASS-ON-THE-EDGE-OF-THE-BAR-NEXT-TO-HIS-ELBOW-
AND-WAIT-UNTIL-YOUR-DRINK-GOES-FLYING-OFF] 0.64000005
```

In this example, the user, Ann, is not at an optimal age or weight for her height. However, she does have working for her the blue eyes, blond hair, and heels. So overall she is considered average in terms of attractiveness. Her profession in education indicates that she is experienced in working with people and is likely to be extroverted. Additionally, as an educator who wears loose clothing to go out at night, she is probably also reserved and not really creative. The man she is targeting is well dressed so he is probably respectable and wealthy. From the observation that he is talking loudly and gesturing wildly, one might infer that he is drunk as well. Given these intermediate conclusions, an attractive or daring woman would be able to easily take advantage of the prospective man's lack of company and his qualities. However, since Ann is reserved and only averagely attractive, she does not have as many options, and the approaches that will give her the greatest chance of success would take advantage of the conscience of the bartender or the man by tricking one into thinking he has spilled her drink.

This example characterizes the type of problems the free drinks advisor system can handle. Our system is capable of taking objective information that the user should know or be able to observe and through the process of elimination and inferences, find successful paths to viable solutions. One key point to make is that the scope of our problem is recommending plausible approaches to receiving a free drink, given the relevant input information about the woman, man, and venue. The recommendations are useful given that the user has inputted correct information without fabricating any answers, such as her weight, and that it is her first time meeting the relevant parties involved in the scenario. The problems of trying to get several drinks from one man or from a man she already knows are both out of scope for the functionalities of our system. An obvious example is that if a woman succeeds the first time by stealing a man's drink, reasonably if he noticed, the second time he would not allow that to happen again. This type of problem only differs from the problem in scope slightly, but due to the volatile nature of human thoughts on which the success of our advisor system depends, it becomes out of scope.

## 2. Knowledge Representation

We have chosen a rule-based knowledge representation since it seems to best match the scope of the problem we aimed to solve. In particular, rules provide the advantages of an easily extensible knowledge base, modularity of chunks of knowledge, and the ease of categorization of associations.

Our knowledge base includes both factual knowledge and heuristic knowledge. An example rule of factual knowledge is the calculation of the Body Mass Index from the user's height and weight, and an example rule of heuristic knowledge is the judgmental assumption that a woman working in the design industry is extroverted. Given the scope of our problem, which involves an immense amount of variables and perceptions, it is no surprise that an overwhelming majority of the knowledge is heuristic. This in turn means that the knowledge base has much potential to be expanded in order to account for more of the variables and to capture more precise knowledge. Thus, it is important that our knowledge representation permits easy ways of expanding and modifying the knowledge base. As we learned in class, rule's independent nature provides this flexibility.

Next, as briefly discussed in the task description, the knowledge of the system captures two main types of knowledge: (1) association between observable attributes and intermediate qualities such as attractiveness; and (2) association between intermediate qualities and recommendations. An example rule that represent type (1) dictates that if the woman wears tight clothing or if her profession is administrative, marketing, education, design, or business, then she is an extrovert (see Rule 1). An example rule that represent type (2) knowledge indicates that if the woman is attractive, extroverted, and creative, the venue is loud, and the man is respectable, wealthy, and alone, she can try to get a free drink by making up a sob story to the man (see Rule 2). These two knowledge types form the main tiers between inputs and the result. Furthermore, there are four fairly independent main categories to which an association can belong: (1) woman's attractiveness; (2) woman's personality traits; (3) man's behavior and state; (4) venue characteristics . One way to visualize such a knowledge base is as a hierarchical tree structure (see figure 1 in the next section), in which the basic attributes are the leaves and each association captured by a rule provides the link from a node of a lower level in the tree to a node in the next level. At the root of the tree is the desired outcome, which is the recommended course of action. Each branch from the root is a category. This visual representation of the knowledge base will be discussed in more details in the next two sections. Meanwhile, keeping this hierarchical and categorical structure in mind, since the purpose of rules is to specify necessary and sufficient conditions for something to be part of a certain category, we decided that rules provides precisely the type of representation we need for effective encapsulation of stereotypical associations of qualities about the woman, man, setting, and course of action.

**Rule 1**

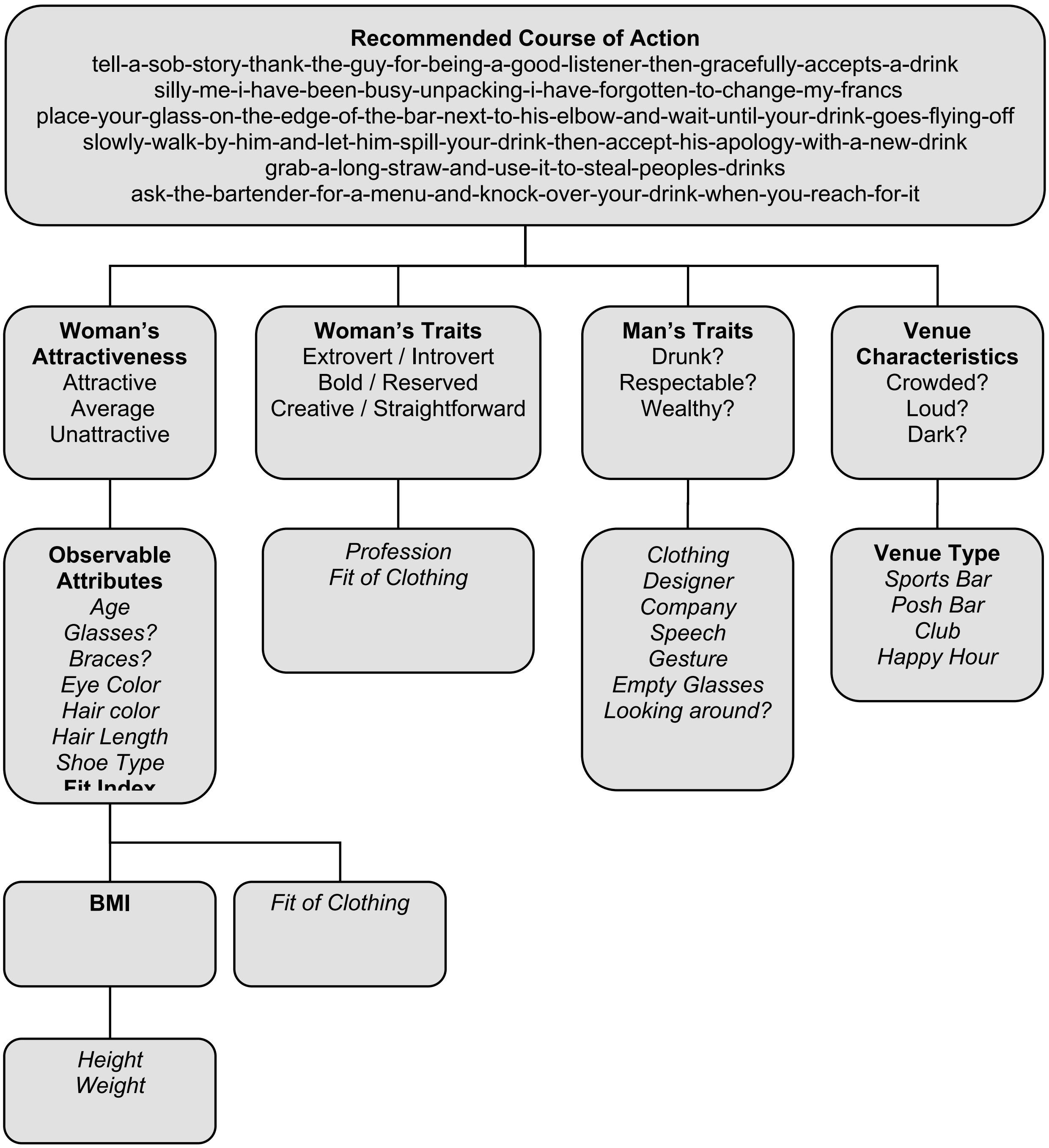
```
(defrule woman-extrovert (:backward :certainty 1.0)
  if [or [fit ?who tight]
         [or [job ?who administrative]
              [job ?who marketing]
              [job ?who education]
              [job ?who design]
              [job ?who business]]]
  then [woman-trait1 ?who extrovert])
```

**Rule 2**

```
(defrule tell-a-sob-story (:backward :certainty 1.0 :importance 99)
  if [and [attractiveness ?who attractive]
             [woman-trait1 ?who extrovert]
             [woman-trait2 ?who creative]
         [venue-loud ?who no]
         [man-respectable ?who yes]
         [man-wealthy ?who yes]
         [man-alone ?who yes]]
  then [action ?who tell-a-sob-story-thank-the-guy-for-being-a-good-
  listener-then-gracefully-accepts-a-drink])
```

**4. Knowledge Tree**

This tree diagram on the next page represents the general structure and the important content of the knowledge in our free drinks advisor system. The tree branches from the root show the four major categories of intermediate qualities. In each box, the bold title at the top represent the category of the attributes in the same box. The normal phrases or words on each row in the box represent the different values a quality or in Joshua's term, a "class" or "predicate," might take on. The italicized attributes are observable information the system might ask the user.



The node on a higher level can be concluded from the attributes outlined in the boxes to which it has a link on the lower level. Thus chains of reasoning are formed from the goal node to leaf nodes. When no more inferences and back chaining can be performed, the system outputs each action goal with complete chains of reasoning for all its preconditions as a plausible approach for getting a free drink.

**3. Problem Solving Paradigm**

We intend to use backward chaining with our rule-based knowledge base as our problem-solving paradigm. Backward chaining was a natural choice since we want the user to inquire the system for an end result and the system in response should back chain from the goal, ask for relevant information, and eventually find successful paths to the goal.

The inference engine follows a strategy:

Upon reception of an inquiry, the system tries out each possible backward rule in the knowledge base that has as a consequent a recommended course of action in the order of rated importance. It looks at the preconditions of the action in the IF clause of the rule and asks each predication. If the asked predication is the consequent of other rules, this backward chaining and predicate asking continues to perpetrate until the predication is not a consequent of any rule, at which point the system asks the user to answer the question. The same perpetration of backward reasoning occurs through each relevant branch as identified by the preconditions until all required preconditions have been either determined to be failed or successful. Our system is designed such that it offers all the plausible recommendations to the user. Therefore, even if one backward rule with action as its consequent succeeds, the system continues to perform the same backward chaining for all the possible actions. The actions that succeed will in essence have established a successful line of reasoning from the root of the knowledge tree down to all the leaves that determine the intermediate predications. When the system quits the last action backward rule, it will output all of the action predicates that were found to be successful. The predications will be listed depending on the degree of stringency of required qualities for their success. This strategy shall become clear in the annotated example of user-system dialogue.

The order of outputted recommendations is determined by the intuition that attractiveness is the main factor influencing the options of the woman in getting a free drink. An attractive woman, of course, has more options than her less attractive counterparts and can probably get men fighting to buy her drinks without any effort. The three remaining categories, woman's personality, man's traits, and setting, are then used to determine which of the remaining course of actions are appropriate and in what order of appropriateness. For example, if the venue is dark and crowded, it would be more likely for a woman to succeed in stealing people's drinks. Finally, if no course of action succeeds, we have included in our system a fallback plan that any woman can try and should be able to pull off.

## 4. Implementation Logistics

Since we will be using a rule-based representation, Joshua seems to be the best tool for our program. Through the last problem set we have gained experience in running and writing rules with Joshua. Although the interface is somewhat primitive, the implementation of rules is straightforward and it is also the preferred tool for projects as indicated on the course website.

As for the actual implementation, since our knowledge is primarily organized into categories that are fairly independent of one another, we decided to take advantage of the modularity of the categorical inferences and divide up the implementation of rules by category.

## 5. Annotated Example

At the beginning of this example, I will annotate with more technical details in order to fully illustrate the back chaining and inference reasoning of the system. Later on I will provide more descriptive explanations for the actions of the system.

```
(ask [action tina ?x] #'suggest)
```

*> System tries matching predication [ACTION TINA ?X], which is not in the database, so it asks the predication. The first found matching rule with its conclusion matching [ACTION TINA VALUE] is rule TELL-A-SOB-STORY. First precondition is [ATTRACTIVENESS TINA ATTRACTIVE], so the system needs to determine whether TINA is ATTRACTIVE. This predication is the conclusion of rule WOMAN-ATTRACTIVE, so system continues back chaining. First precondition is [AGE-INDEX TINA ?A], which means the first factor required to determine ATTRACTIVENESS is AGE. The first rule found with that as conclusion is AGE-INDEX-3. The precondition of this rule matches [AGE TINA ?X], which is not a conclusion of any rule and database has no knowledge on TINA's AGE, so the system asks the user for information.*

*>> This is a fully detailed description of the back chaining inference engine at work to figure out that it needs to ask for TINA's age. For the next sequence of questions, the system is trying to determine whether TINA is ATTRACTIVE. Each criterion is assigned an index score, with 3 being the most attractive, 2 being average, and 1 being the least attractive quality of that certain attribute.*

```
What is TINA's age: 23
```

*> [AGE TINA 23] is inserted into the database and systems checks rules AGE-INDEX-2 and AGE-INDEX-1 since they both satisfy [AGE-INDEX TINA ?A] from WOMAN-ATTRACTIVE. The preconditions of these rules matches [AGE TINA 23] so no more questions are asked. Only AGE-INDEX-3 is satisfied, so [AGE-INDEX TINA 3] enters database. Next precondition that determine WOMAN-ATTRACTIVE is [BMI-INDEX TINA ?B], meaning BMI-INDEX is also required for determining ATTRACTIVENESS. Thus system tries first found rule, BMI-INDEX-3, which requires [BODY-TYPE TINA IDEAL]. Database has no predication with class BODY-TYPE, so it tries the backward rule BODY-TYPE-ANOREXIC, which depends on [HEIGHT TINA ?X]*

*and [WEIGHT TINA ?Y]. The back chain stops here, so the system asks the user.*

What is TINA's height: 66

What is TINA's weight: 125

*> Predication [HEIGHT TINA 66] and [WEIGHT TINA 125] are inserted into database. System checks all the BODY-TYPE rules and backward rule BODY-TYPE-IDEAL succeeds, thus [BODY-TYPE TINA IDEAL] enters database. Going 1 step back up the tree, BMI-INDEX rules are checked to see which one can be satisfied by this BODY-TYPE. [BMI-INDEX TINA 3] succeeds and enters database. The third precondition in WOMAN-ATTRACTIVE is [GLASSES-INDEX TINA ?C], so system tries GLASS-INDEX rules. They require knowledge on [WEARS-GLASSES TINA NO], so the system asks the user.*

Is it the case that TINA wears glasses: No

*> System enters [WEARS-GLASSES TINA NO] and [GLASSES-INDEX TINA 3] following similar logic as how system entered [BODY-TYPE TINA IDEAL] and [BMI-INDEX TINA 3]. The rest of the preconditions for WOMAN-ATTRACTIVE involve BRACES, EYE-COLOR, HAIR-LENGTH, SHOE-TYPE, FIT-OF-CLOTHING, and SKIRT-LENGTH so the system back chains in similar manner through associated rules for each one and asks the user questions.*

Is it the case that TINA has braces: No

*> Enter [HAS-BRACES TINA NO] and [BRACES-INDEX TINA 3]*

Is it the case that TINA has blue eyes: Yes

*> Enter [EYE-COLOR-BLUE TINA YES] and [EYE-COLOR-INDEX TINA 3]*

Is it the case that TINA has blonde hair: No

*> Enter [HAIR-COLOR-BLONDE TINA NO] and [HAIR-COLOR-INDEX TINA 2]*

What is TINA's hair length: Medium

*> Enter [HAIR-LENGTH TINA MEDIUM] and [HAIR-LENGTH-INDEX TINA 2]*

What is TINA's shoe type: Heels

*> Enter [SHOE-TYPE TINA HEELS] and [SHOE-TYPE-INDEX TINA 3]*

What is TINA's clothing fit: Tight

*> Enter [FIT TINA TIGHT] and [FIT-OF-CLOTHING-INDEX TINA 3]. The FIT-OF-CLOTHING-INDEX requires one further level of back chaining than the above few questions since it depends on both BMI-INDEX and FIT.*

Is it the case that TINA is wearing a skirt: Yes

*> Enter [WEARS-SKIRT TINA YES]. The system design gives bonus attractiveness points to women that wear skirts, and the shorter the higher the SKIRT-INDEX she will receive. So it first determines whether she's wearing a skirt. An index score of 0 is assigned to a woman who is not wearing a skirt. If she is, then the system inquires about the length.*

What is TINA's skirt length: Medium

*> Enter [SKIRT-LENGTH TINA MEDIUM]*

*> At the end of this series of question, the system has back chained WOMAN-ATTRACTIVE to all the user-input attributes and collected all required information to determine that TINA is ATTRACTIVE and enters in the predication [ATTRACTIVENESS TINA ATTRACTIVE]*

*> Now the system is still within the trying of the backward rule TELL-A-SOB-STORY, and since ATTRACTIVENESS has been determined, now it tries the rest of the preconditions. The next series of questions are asked in order to determine whether TELL-A-SOB-STORY is a plausible approach. The next required precondition, which is [WOMAN-TRAIT1 TINA EXTROVERT]. EXTROVERT depends on FIT and JOB, so system asks about profession.*

What is TINA's profession: Marketing

*> Enter [JOB TINA MARKETING] and [WOMAN-TRAIT1 TINA EXTROVERT] since the preconditions for being an EXTROVERT was satisfied.*

*> All tried precondition for TELL-A-SOB-STORY is satisfied so far, so system tries the next one, [WOMAN-TRAIT2 TINA CREATIVE], which only depends on profession. Thus no additional question was asked to the user.*

*> The next precondition is [VENUE-LOUD TINA NO], so that TINA can have a conversation with the man and tell a story that will spark sympathy. System back chains down the tree and checks rule VENUE-LOUD to see if TINA's at a LOUD place. Database has no information on VENUE, so system asks user.*

What is TINA's location(venue type): Happy-Hour

*> Enter [VENUE TINA HAPPY-HOUR] and [VENUE-LOUD TINA NO] since HAPPY-HOUR is classified as a venue not so loud that a conversation cannot take place. TELL-A-SOB-STORY also requires the MAN to be RESPECTABLE, which depends on whether he is well dressed. System asks the user for that observed information.*

Is it the case that TINA's target man is wearing a suit: Yes

*> Enter [MAN-SUIT TINA YES] and [MAN-RESPECTABLE TINA YES]*

*> Next, a WEALTHY MAN is also required, so system tries backward rule to ask [MAN-WEALTHY TINA YES]. Since WEALTHY trait can be inferred from being RESPECTABLE, system needs no further questions to determine that MAN is WEALTHY and add [MAN-WEALTHY TINA YES] to database.*

*> So far al preconditions for SOB-STORY are satisfied, the last one to check is whether MAN is ALONE. There is no backward rule to try, so it is a direct question.*

Is it the case that TINA's target man is alone: Yes

*> Enter [MAN-ALONE TINA YES] and now TELL-A-SOB-STORY has completed its chain of reasoning and [ACTION TINA TELL-A-SOB-STORY-THANK-THE-GUY-FOR-BEING-A-GOOD-LISTENER-THEN-GRACEFULLY-ACCEPTS-A-DRINK] is entered into the database.*

*>> Now that SOB-STORY is successful, the system exits the rule and is back to trying to find more matches in the conclusion of rules for the original inquiry, [ACTION TINA VALUE]. It tries all the rules inferring the recommended actions and the next two in order, SILLY-ME and KLUTZ-1, are satisfied by all the predications already justified in the database without having to ask more questions to the user. Thus [ACTION TINA SILLY-ME-I-HAVE-BEEN-BUSY-UNPACKING-I-HAVE-FORGOTTEN-TO-*

*CHANGE-MY-FRANCS] and [ACTION TINA PLACE-YOUR-GLASS-ON-THE-EDGE-OF-THE-BAR-NEXT-TO-HIS-ELBOW-AND-WAIT-UNTIL-YOUR-DRINK-GOES-FLYING-OFF] are entered as well.*

*>> However, when the system tries the backward rule I-AM-SUCH-A-KLUTZ-2, it does not know whether the MAN is DRUNK, thus it back chains to ask the predication [MAN-DRUNK TINA YES]. The precondition to being DRUNK is either if the NUMBER-EMPTY-GLASSES is greater than 4 or if MAN is SPEAKING-LOUDLY and MAKING-LARGE-GESTURES. System first tries to find out about NUMBER-EMPTY-GLASSES.*

What is TINA's target man's number of empty glasses in front him: 5

*> [NUMBER-EMPTY-GLASSES TINA 5] is entered and that by itself satisfies [MAN-DRUNK TINA YES] without finding out about SPEAKING-LOUDLY and MAKING-GESTURES. Now all the preconditions for I-AM-SUCH-A-KLUTZ-2 have been satisfied as well, so system adds [ACTION TINA SLOWLY-WALK-BY-HIM-AND-LET-HIM-SPILL-YOUR-DRINK-THEN-ACCEPT-HIS-APOLOGY-WITH-A-NEW-DRINK] to database.*

*>> Now system continues trying the other ACTIONS. SECRET-AGENT-STEALTH-SIPPER is not plausible because it requires VENUE to be DARK, but back chaining to VENUE-DARK rule confirms that HAPPY-HOUR is NOT-DARK.*
*>> Finally, the last and the default fall back ACTION, I-AM-SUCH-A-KLUTZ-3, is satisfied and added to the database. This ACTION requires the WOMAN to be ATTRACTIVE, AVERAGE, or UNATTRACTIVE, which applies to every user, thus the program will at least always give one option for the user. Thus [ACTION TINA ASK-THE-BARTENDER-FOR-A-MENU-AND-KNOCK-OVER-YO*
*UR-DRINK-WHEN-YOU-REACH-FOR-IT] is also added to the database.*

*>> Now that no more inferences or chaining can be made, the system respond to the original user ASK for her successful [ACTION TINA ?x], and outputs all the matching predications that have been inserted into the database during the process.*

[ACTION TINA ASK-THE-BARTENDER-FOR-A-MENU-AND-KNOCK-OVER-YOUR-DRINK-WHEN-YOU-REACH-FOR-IT] 1.0

[ACTION TINA SLOWLY-WALK-BY-HIM-AND-LET-HIM-SPILL-YOUR-DRINK-THEN-ACCEPT-HIS-APO
LOGY-WITH-A-NEW-DRINK] 0.8

[ACTION TINA PLACE-YOUR-GLASS-ON-THE-EDGE-OF-THE-BAR-NEXT-TO-HIS-ELBOW-AND-WAIT-UNTIL-YOUR-DRINK-GOES-FLYING-OFF] 0.64000005

[ACTION TINA SILLY-ME-I-HAVE-BEEN-BUSY-UNPACKING-I-HAVE-FORGOTTEN-TO-CHANGE-MY-FRANCS] 0.8

[ACTION TINA TELL-A-SOB-STORY-THANK-THE-GUY-FOR-BEING-A-GOOD-LISTENER-THEN-GRACEFULLY-ACCEPTS-A-DRINK] 0.64000005

By contrast to the example of use Ann given in the beginning of the paper, Tina has five options! This is because Tina is attractive, bold, and creative, and the man she is targeting is drunk, respective, and wealthy. Therefore she can fully take advantage of her own desirable and useful qualities and of the benign and vulnerable state of the target man, in addition to having the pretend klutzy options that are available for most women.

The resulting database from Tina's inquiry:

```
True things
[ATTRACTIVENESS TINA ATTRACTIVE]
[FIT-OF-CLOTHING-INDEX TINA 3]
[SKIRT-LENGTH-INDEX TINA 2]
[SHOE-TYPE-INDEX TINA 3]
[WOMAN-TRAIT3 TINA BOLD]
[VENUE TINA HAPPY-HOUR]
[HAIR-LENGTH-INDEX TINA 2]
[HAIR-COLOR-BLONDE TINA NO]
[WOMAN-TRAIT1 TINA EXTROVERT]
[JOB TINA MARKETING]
[AGE TINA 23]
[MAN-DRUNK TINA YES]
[MAN-ALONE TINA YES]
[VENUE-DARK TINA NO]
[AGE-INDEX TINA 3]
[HAS-BRACES TINA NO]
[EYE-COLOR-INDEX TINA 3]
[SHOE-TYPE TINA HEELS]
[VENUE-CROWDED TINA YES]
[ACTION TINA ASK-THE-BARTENDER-FOR-A-MENU-AND-KNOCK-OVER-YOUR-DRINK-
WHEN-YOU-REACH-FOR-IT]
[ACTION TINA SLOWLY-WALK-BY-HIM-AND-LET-HIM-SPILL-YOUR-DRINK-THEN-
ACCEPT-HIS-APOLOGY-WITH-A-NEW-DRINK]
[ACTION TINA PLACE-YOUR-GLASS-ON-THE-EDGE-OF-THE-BAR-NEXT-TO-HIS-ELBOW-
AND-WAIT-UNTIL-YOUR-DRINK-GOES-FLYING-OFF]
[ACTION TINA SILLY-ME-I-HAVE-BEEN-BUSY-UNPACKING-I-HAVE-FORGOTTEN-TO-
CHANGE-MY-FRANCS]
[ACTION TINA TELL-A-SOB-STORY-THANK-THE-GUY-FOR-BEING-A-GOOD-LISTENER-
THEN-GRACEFULLY-ACCEPTS-A-DRINK]
[MAN-RESPECTABLE TINA YES]
[WEARS-SKIRT TINA YES]
[VENUE-LOUD TINA NO]
[WEARS-GLASSES TINA NO]
[WOMAN-TRAIT2 TINA CREATIVE]
[MAN-WEALTHY TINA YES]
[HEIGHT TINA 66]
[EYE-COLOR-BLUE TINA YES]
[BMI-INDEX TINA 3]
[MAN-SUIT TINA YES]
[NUMBER-EMPTY-GLASSES TINA 5]
[BRACES-INDEX TINA 3]
[HAIR-COLOR-INDEX TINA 2]
[HAIR-LENGTH TINA MEDIUM]
[FIT TINA TIGHT]
[WEIGHT TINA 125]
[GLASSES-INDEX TINA 3]
[SKIRT-LENGTH TINA MEDIUM]
[BODY-TYPE TINA IDEAL]
False things
```

**6. Reflections**

Overall, our system encapsulated some basic knowledge about women's attractiveness, personality traits, men's behaviors, and night scene venues and performs fairly well in choosing probable recommendations of getting a free drink depending on the user's particular scenario.

The rule-based knowledge representation served our purpose well in several ways. First of all, rules provided a simple but effective way of capturing the heuristic knowledge that enabled inferences from observable objective information to result in recommendations based on stereotypical assumptions. More specifically, the rule-based backward chaining permitted inferences from observable physical attributes to a rating of attractiveness, from profession and fit of clothing to personality traits, from behaviors and dress to qualities and expected actions, and from venue types to setting characteristics. Secondly, the independent nature of rules made it possible to encode each association as a separate piece of knowledge such that they could be easily categorized into larger units with other related rules. This advantage made it easy to divide up the implementation efforts and to organize our knowledge into a coherent tree structure composed of categories. Thirdly, as we learned in class and experienced during the design and implementation of our program, rules definitely is a representation that allows easy expansion and modification of the knowledge base.

In terms of implementation, our chosen tool, Joshua, although primitive, provided sufficient functionalities to encode the knowledge and to aid us in debugging. The syntax of Joshua code, upon viewing, is easy to understand and resembles seduced. Given the type of associations we wanted to capture, the IF-THEN structure proved to be an intuitive way of implementation. The "tracing all" and "show database" options were also very helpful in deciphering what the inference engine was doing and in ensuring the correct inferences were being made.

However, as all programs, our system had its limitations. Although our system captured of the basic stereotypical assumptions as knowledge to make a plausible suggestion to the user, we realized that that reasoning about human personalities, behaviors, and judgments is an extremely difficult problem. Although there are definitely certain rules of thumb that are commonly accepted, such as that women that wear glasses are not as attractive, I personally know men that have an appreciation for the power woman or nerdy librarian type looks. This is just one example to illustrate the endless possibilities that the program may encounter depending on the woman, man, and setting. One way we considered to make the knowledge more precise was to place different weights on different attributes. For example, in most cases, a woman's attractiveness to a man would depend more on her body type than on her eye color. However, it was hard to encode corresponding weights to the attributes in a natural way with the rule representation. We solved this problem by taking an alternative approach: for each physical attribute used to infer attractiveness, an index indicating an attractiveness level was assigned, and the final attractiveness assignment was made based on a composite score. This approach worked well for our problem, however, it was somewhat awkward and roundabout.

Another issue we faced in both the design and testing phases was the uncertainty of the accuracy of the outputted results. Whereas a medical diagnosis program such as MYCIN, which if given symptoms of a disease from a textbook, will give a diagnosis that one can determine to be the "right" end result given the inputs, our free drinks advisor, face immense uncertainty as two women with the exact same inputs may not become successful with the same approach. For example, two women might both try to steal other people's drinks with a long straw, but it may happen that one woman succeeds and the woman with the shorter arm fails with that approach. Our solution to the problem was to output all of the plausible recommendations in order of difficulty to pull off. This way, if a woman knows that she has short arms, she might opt to use an approach other than the stealing drink one.

While working on this project, I am now enlightened as to why the topic knowledge based systems is such an interesting area of study with much room for further exploration although such substantial work has already been done. I have personally experienced the difficulty of capturing expert knowledge and trying to emulate the associations of real world scenarios.

## 7. Appendix

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; RULES ABOUT: woman's physical attractiveness ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;; RULES ABOUT: age index

(defrule age-index-3 (:backward :certainty 1.0)
  if [and [age ?who ?x]
          (<= ?x 30)]
  then [age-index ?who 3])

(defrule age-index-2 (:backward :certainty 1.0)
  if [and [age ?who ?x]
          (> ?x 30)
          (<= ?x 40)]
  then [age-index ?who 2])

(defrule age-index-1 (:backward :certainty 1.0)
  if [and [age ?who ?x]
          (> ?x 40)]
  then [age-index ?who 1])

;;; RULES ABOUT: body type

(defrule body-type-anorexic (:backward :certainty 1.0)
  if [and [height ?who ?x]
          [weight ?who ?y]
          (< (/ (* ?y 0.4536) (* (* ?x 0.0254) (* ?x 0.0254))) 17)]
  then [body-type ?who anorexic])

(defrule body-type-underweight (:backward :certainty 1.0)
  if [and [height ?who ?x]
          [weight ?who ?y]
          (>= (/ (* ?y 0.4536) (* (* ?x 0.0254) (* ?x 0.0254))) 17)
        (< (/ (* ?y 0.4536) (* (* ?x 0.0254) (* ?x 0.0254))) 19)]
  then [body-type ?who underweight])

(defrule body-type-ideal (:backward :certainty 1.0)
  if [and [height ?who ?x]
          [weight ?who ?y]
          (>= (/ (* ?y 0.4536) (* (* ?x 0.0254) (* ?x 0.0254))) 19)
        (< (/ (* ?y 0.4536) (* (* ?x 0.0254) (* ?x 0.0254))) 26)]
  then [body-type ?who ideal])

(defrule body-type-overweight (:backward :certainty 1.0)
  if [and [height ?who ?x]
          [weight ?who ?y]
          (>= (/ (* ?y 0.4536) (* (* ?x 0.0254) (* ?x 0.0254))) 26)
        (< (/ (* ?y 0.4536) (* (* ?x 0.0254) (* ?x 0.0254))) 28)]
  then [body-type ?who underweight])

(defrule body-type-obese (:backward :certainty 1.0)
  if [and [height ?who ?x]
          [weight ?who ?y]
```

```
          (>= (/ (* ?y 0.4536) (* (* ?x 0.0254) (* ?x 0.0254))) 28)]
  then [body-type ?who obese])

;;; RULES ABOUT: BMI index

(defrule bmi-index-3 (:backward :certainty 1.0)
  if [body-type ?who ideal]
  then [bmi-index ?who 3])

(defrule bmi-index-2 (:backward :certainty 1.0)
  if [or [body-type ?who underweight]
         [body-type ?who overweight]]
  then [bmi-index ?who 2])

(defrule bmi-index-1 (:backward :certainty 1.0)
  if [or [body-type ?who anorexic]
         [body-type ?who obese]]
  then [bmi-index ?who 1])

;;; RULES ABOUT: wearing glasses

(defrule glasses-index-3 (:backward :certainty 1.0)
  if [wears-glasses ?who no]
  then [glasses-index ?who 3])

(defrule glasses-index-1 (:backward :certainty 1.0)
  if [wears-glasses ?who yes]
  then [glasses-index ?who 1])

;;; RULES ABOUT: has braces

(defrule braces-index-3 (:backward :certainty 1.0)
  if [has-braces ?who no]
  then [braces-index ?who 3])

(defrule braces-index-1 (:backward :certainty 1.0)
  if [has-braces ?who yes]
  then [braces-index ?who 1])

;;; RULES ABOUT: eye color

(defrule eye-color-index-3 (:backward :certainty 1.0)
  if [eye-color-blue ?who yes]
  then [eye-color-index ?who 3])

(defrule eye-color-index-2 (:backward :certainty 1.0)
  if [eye-color-blue ?who no]
  then [eye-color-index ?who 2])

;;; RULES ABOUT: hair color

(defrule hair-color-index-3 (:backward :certainty 1.0)
  if [hair-color-blonde ?who yes]
  then [hair-color-index ?who 3])

(defrule hair-color-index-2 (:backward :certainty 1.0)
  if [hair-color-blonde ?who no]
```

```
  then [hair-color-index ?who 2])

;;; RULES ABOUT: hair length

(defrule hair-length-index-3 (:backward :certainty 1.0)
  if [hair-length ?who long]
  then [hair-length-index ?who 3])

(defrule hair-length-index-2 (:backward :certainty 1.0)
  if [hair-length ?who medium]
  then [hair-length-index ?who 2])

(defrule hair-length-index-1 (:backward :certainty 1.0)
  if [hair-length ?who short]
  then [hair-length-index ?who 1])

;;; RULES ABOUT: shoe type

(defrule shoe-type-index-3 (:backward :certainty 1.0)
  if [shoe-type ?who heels]
  then [shoe-type-index ?who 3])

(defrule shoe-type-index-2 (:backward :certainty 1.0)
  if [shoe-type ?who flats]
  then [shoe-type-index ?who 2])

(defrule shoe-type-index-1 (:backward :certainty 1.0)
  if [shoe-type ?who sneakers]
  then [shoe-type-index ?who 1])

;;; RULES ABOUT: fit of clothing

(defrule fit-of-clothing-index-3 (:backward :certainty 1.0)
  if [and [bmi-index ?who 3]
          [or [fit ?who tight]
              [fit ?who fitted]]]
  then [fit-of-clothing-index ?who 3])

(defrule fit-of-clothing-index-ideal-2 (:backward :certainty 1.0)
  if [and [bmi-index ?who 3]
          [fit ?who loose]]
  then [fit-of-clothing-index ?who 2])

(defrule fit-of-clothing-index-2 (:backward :certainty 1.0)
  if [and [bmi-index ?who ?x]
          (< ?x 3)
          [fit ?who fitted]]
  then [fit-of-clothing-index ?who 2])

(defrule fit-of-clothing-index-1 (:backward :certainty 1.0)
  if [and [bmi-index ?who ?x]
          (< ?x 3)
          [fit ?who loose]]
  then [fit-of-clothing-index ?who 1])

(defrule fit-of-clothing-index-0 (:backward :certainty 1.0)
  if [and [bmi-index ?who ?x]
```

```
          (< ?x 3)
          [fit ?who tight]]
  then [fit-of-clothing-index ?who 0])

;;; RULES ABOUT: skirt length

(defrule skirt-length-index-0 (:backward :certainty 1.0)
  if [wears-skirt ?who no]
  then [skirt-length-index ?who 0])

(defrule skirt-length-index-3 (:backward :certainty 1.0)
  if [and [wears-skirt ?who yes]
          [skirt-length ?who short]]
  then [skirt-length-index ?who 3])

(defrule skirt-length-index-2 (:backward :certainty 1.0)
  if [and [wears-skirt ?who yes]
          [skirt-length ?who medium]]
  then [skirt-length-index ?who 2])

(defrule skirt-length-index-1 (:backward :certainty 1.0)
  if [and [wears-skirt ?who yes]
          [skirt-length ?who long]]
  then [skirt-length-index ?who 1])

;;; RULES ABOUT: woman's physical attractiveness

(defrule woman-attractive (:backward :certainty 1.0)
  if [and [age-index ?who ?a]
          [bmi-index ?who ?b]
        [glasses-index ?who ?c]
        [braces-index ?who ?d]
        [eye-color-index ?who ?e]
        [hair-color-index ?who ?f]
        [hair-length-index ?who ?g]
        [shoe-type-index ?who ?h]
        [fit-of-clothing-index ?who ?i]
        [skirt-length-index ?who ?j]
        (> (+ ?a ?b ?c ?d ?e ?f ?g ?h ?i ?j) 21)]
  then [attractiveness ?who attractive])

(defrule woman-average (:backward :certainty 1.0)
  if [and [age-index ?who ?a]
          [bmi-index ?who ?b]
        [glasses-index ?who ?c]
        [braces-index ?who ?d]
        [eye-color-index ?who ?e]
        [hair-color-index ?who ?f]
        [hair-length-index ?who ?g]
        [shoe-type-index ?who ?h]
        [fit-of-clothing-index ?who ?i]
        [skirt-length-index ?who ?j]
        (<= (+ ?a ?b ?c ?d ?e ?f ?g ?h ?i ?j) 21)
        (> (+ ?a ?b ?c ?d ?e ?f ?g ?h ?i ?j) 15)]
  then [attractiveness ?who average])

(defrule woman-unattractive (:backward :certainty 1.0)
```

```
  if [and [age-index ?who ?a]
          [bmi-index ?who ?b]
        [glasses-index ?who ?c]
        [braces-index ?who ?d]
        [eye-color-index ?who ?e]
        [hair-color-index ?who ?f]
        [hair-length-index ?who ?g]
        [shoe-type-index ?who ?h]
        [fit-of-clothing-index ?who ?i]
        [skirt-length-index ?who ?j]
        (<= (+ ?a ?b ?c ?d ?e ?f ?g ?h ?i ?j) 15)]
  then [attractiveness ?who unattractive])

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; RULES ABOUT: woman's personality traits ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;; RULES ABOUT: extrovert or introvert

(defrule woman-extrovert (:backward :certainty 1.0 :importance 70)
  if [or [fit ?who tight]
         [job ?who adminstrative]
         [job ?who marketing]
         [job ?who education]
         [job ?who design]
         [job ?who business]]
  then [woman-trait1 ?who extrovert])

(defrule woman-introvert (:backward :certainty 0.8 :importance 69)
  if [and [or [fit ?who fitted] [fit ?who loose]]
          [or [job ?who sciences] [job ?who technology]]]
  then [woman-trait1 ?who introvert])

;;; RULES ABOUT: creative or straightforward

(defrule woman-creative (:backward :certainty 0.8 :importance 68)
  if [or [job ?who design]
         [job ?who marketing]]
  then [woman-trait2 ?who creative])

(defrule woman-straightforward (:backward :certainty 0.9 :importance
67)
  if [or [job ?who adminstrative]
         [job ?who sciences]
         [job ?who education]
         [job ?who technology]
         [job ?who business]]
  then [woman-trait2 ?who straightforward])

;;; RULES ABOUT: reserved or bold

(defrule woman-reserved (:backward :certainty 0.8 :importance 66)
  if [or [or [fit ?who fitted] [fit ?who loose]]
         [job ?who administrative]
         [job ?who education]
         [job ?who sciences]
         [job ?who technology]]
```

```
  then [woman-trait3 ?who reserved])

(defrule woman-bold (:backward :certainty 0.9 :importance 65)
  if [or [fit ?who tight]
       [job ?who business]
         [job ?who design]
         [job ?who marketing]]
  then [woman-trait3 ?who bold])

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; RULES ABOUT: venue characteristics ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;; RULES ABOUT: whether venue's crowded

(defrule venue-crowded (:backward :certainty 0.8)
  if   [or [venue ?who sports-bar]
           [venue ?who club]
           [venue ?who happy-hour]]
  then [venue-crowded ?who yes])

(defrule venue-not-crowded (:backward :certainty 0.8)
  if  [venue ?who posh-bar]
  then [venue-crowded ?who no])

;;; RULES ABOUT: whether venue's dark

(defrule venue-dark (:backward :certainty 0.8)
  if   [or [venue ?who posh-bar]
           [venue ?who sports-bar]
           [venue ?who club]]
  then [venue-dark ?who yes])

(defrule venue-not-dark (:backward :certainty 0.8)
  if  [venue ?who happy-hour]
  then [venue-dark ?who no])

;;; RULES ABOUT: whether venue's loud

(defrule venue-loud (:backward :certainty 0.8)
  if  [venue ?who club]]
  then [venue-loud ?who yes])

(defrule venue-not-loud (:backward :certainty 0.8)
  if   [or [venue ?who posh-bar]
           [venue ?who happy-hour]
           [venue ?who sports-bar]]
  then [venue-loud ?who no])

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; RULES ABOUT: man's behaviors and traits ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;; RULES ABOUT: whether man's drunk

(defrule man-drunk (:backward :certainty 0.8)
  if [or [and [number-empty-glasses ?who ?x]
```

```
              (> ?x 4)]
       [and [number-empty-glasses ?who ?x]
              (<= ?x 4)
            [speaking-loudly ?who yes]
            [making-large-gestures ?who yes]]]
  then [man-drunk ?who yes])

(defrule man-not-drunk (:backward :certainty 0.9)
  if [not [known [man-drunk ?who ?]]]
  then [man-drunk ?who no])

;;; RULES ABOUT: whether man's respectable

(defrule man-respectable (:backward :certainty 0.8 :importance 60)
  if [or [man-suit ?who yes]
        [and [man-suit ?who no] [man-nice-shirt ?who yes]]]
  then [man-respectable ?who yes])

(defrule man-not-respectable (:backward :certainty 0.8 :importance 50)
  if [and [man-suit ?who no]
         [man-nice-shirt ?who no]]
  then [man-respectable ?who no])

;;; RULES ABOUT: whether man's wealthy

(defrule man-wealthy (:backward :certainty 0.8)
  if [or [man-respectable ?who yes]
        [and [man-respectable ?who no] [man-in-designer-clothes ?who
yes]]]
  then [man-wealthy ?who yes])

(defrule man-not-wealthy (:backward :certainty 0.8)
  if [and [man-respectable ?who no]
         [man-in-designer-clothes ?who no]]
        then [man-wealthy ?who no])

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; RULES ABOUT: suggested course of action ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defrule tell-a-sob-story (:backward :certainty 1.0 :importance 99)
  if [and [attractiveness ?who attractive]
         [woman-trait1 ?who extrovert]
         [woman-trait2 ?who creative]
        [venue-loud ?who no]
        [man-respectable ?who yes]
        [man-wealthy ?who yes]
        [man-alone ?who yes]]
  then [action ?who tell-a-sob-story-thank-the-guy-for-being-a-good-
listener-then-gracefully-accepts-a-drink])

(defrule silly-me (:backward :certainty 1.0 :importance 98)
  if [and [or [attractiveness ?who attractive]
             [attractiveness ?who average]]
         [woman-trait1 ?who extrovert]
         [woman-trait2 ?who creative]
        [venue-loud ?who no]]
```

```
  then [action ?who silly-me-i-have-been-busy-unpacking-i-have-
forgotten-to-change-my-francs])

(defrule i-am-such-a-klutz-1 (:backward :certainty 1.0 :importance 97)
  if [and [venue-crowded ?who yes]
          [man-respectable ?who yes]
          [man-wealthy ?who yes]
        [man-alone ?who yes]]
  then [action ?who place-your-glass-on-the-edge-of-the-bar-next-to-
his-elbow-and-wait-until-your-drink-goes-flying-off])

(defrule i-am-such-a-klutz-2 (:backward :certainty 1.0 :importance 96)
  if [and [venue-crowded ?who yes]
          [man-alone ?who yes]
          [man-drunk ?who yes]]
  then [action ?who slowly-walk-by-him-and-let-him-spill-your-drink-
then-accept-his-apology-with-a-new-drink])

(defrule secret-agent-stealth-sipper (:backward :certainty 1.0
:importance 95)
  if [and [woman-trait3 ?who bold]
          [venue-dark ?who yes]
          [venue-crowded ?who yes]]
  then [action ?who grab-a-long-straw-and-use-it-to-steal-peoples-
drinks])

(defrule i-am-such-a-klutz-3 (:backward :certainty 1.0 :importance 94)
  if [or [attractiveness ?who attractive]
         [attractiveness ?who average]
         [attractiveness ?who unattractive]]
  then [action ?who ask-the-bartender-for-a-menu-and-knock-over-your-
drink-when-you-reach-for-it])
```