

# Documentação

Março de 2012

Este pacote de software contém a implementação de um algoritmo de inferência probabilística em primeira ordem, para utilização em terminologias representadas pela lógica *CRALC*[1].

## 1 Instalação

### Requisitos:

- Java 6 sdk<sup>1</sup>;
- Eclipse IDE for Java developers<sup>2</sup>.

### Procedimentos:

1. Extraia o conteúdo do zip para uma pasta local de seu computador.
2. Abra o Eclipse com o workspace apontando para a pasta extraída no item anterior.
3. Certifique-se que o path do arquivo Proj01\libs\JavaBayes.jar encontre-se nas referências externas do Eclipse.
4. No Eclipse, em “Run Configurations”, execute o arquivo CrALC.java.

## 2 Utilização

O software dispõe de dois modos distintos de utilização, um através de uma interface gráfica, e outro através de prompt de linha de comando para execução em lote. No modo com interface gráfica é exibida a tela do programa, que pode ser vista na Figura 1 enquanto que no modo de linha de comando, os resultados são exibidos em texto na tela, como visto na Figura 2, e salvos em um arquivo de saída. A escolha entre os dois modos é realizada na chamada do programa, de acordo com parâmetros opcionais que podem ser fornecidos. Por default, na ausência de parâmetros, é carregada a interface gráfica, mas a escolha pode ser feita utilizando-se o parâmetro `-gui`:

<sup>1</sup><http://www.oracle.com/br/technologies/java/index.html>

<sup>2</sup><http://www.eclipse.org/>

`[-gui {on|off}]` ativa ou desativa a interface gráfica.

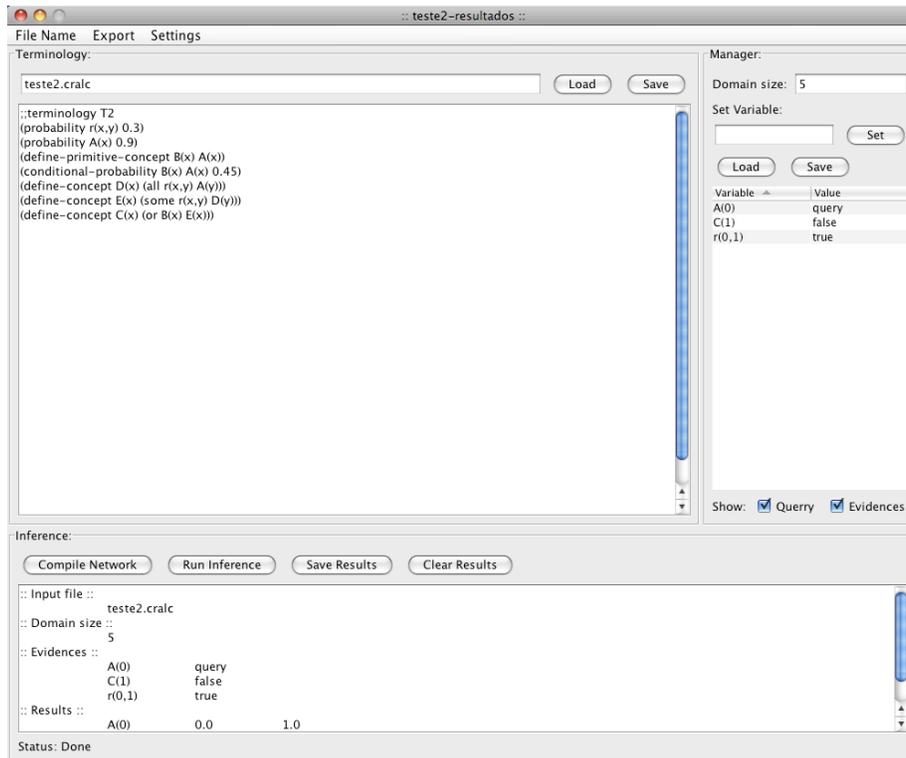


Figura 1: Interface gráfica do pacote de software.

No caso de escolha do modo de linha de comando, mais parâmetros devem ser informados:

`[-i input_file N]` carrega a terminologia especificada num arquivo de entrada e estabelece um domínio de tamanho 'N'.

Alem de informar a terminologia e domínio, no modo de linha de comando, deve utilizar ao menos uma das duas opções a saber: i. exportar uma rede Bayesiana com a proposicionalização da terminologia

`[-p output_file]` proposicionaliza a terminologia e a exporta como uma rede Bayesiana.

ou ii. informar as variáveis da consulta, as variáveis observadas, e um nome para os arquivos dos resultados.

`[-e query_file]` carrega a lista com as variáveis da consulta e com as variáveis observadas.

---

```
Reading input file "teste2.cralc"...
File successfully red
Compiling network
Reading evidence file...
  A(0)  query
  C(1)  false
  r(0,1) true
Evidences successfully red
Generating slices
Running inference... please wait...
Inference took 32ms (17 iterations).
:: Results ::
  A(0) 0.0 1.0
Exporting network: propositionalizing network to teste2-prop.bif
File successfully written
Done
```

---

Figura 2: Resultados na execução em modo de linha de comando.

`[-r name]` salva os resultados da inferência no arquivo `'name.txt'` e salva cada fatia em um arquivo com nome segundo o padrão `name_SliceN.bif`.

Os parâmetros apresentados podem ser utilizados em qualquer ordem, e não só no modo de linha de comando mas também na utilização com interface gráfica. Na interface gráfica, caso os parâmetros não sejam informados na chamada do programa, os campos da interface são exibidos sem nenhum preenchimento; caso contrário os campos são exibidos já preenchidos; em ambos os casos o usuário fica livre para editar e alterar o conteúdo dos campos conforme desejado. A janela do programa é dividida em três regiões principais: superior esquerda, superior direita e inferior; organizadas como visto a seguir.

A região superior esquerda contém um campo para a especificação da terminologia em *CRA $\mathcal{L}\mathcal{C}$*  além de botões para salvar ou carregar a especificação em um arquivo de texto, como pode-se ver na Figura 3. Detalhes sobre o formato desse arquivo serão informados mais a diante, no fim dessa seção.

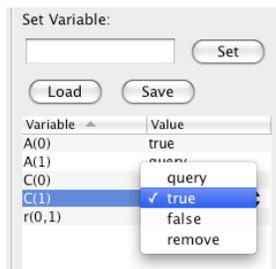
A região superior direita contém um campo para definir o tamanho do domínio a ser utilizado, um campo para adicionar uma variável à lista de observações e consultas, uma tabela contendo a lista de todas as variáveis observadas e todas as consultas, botões para salvar ou carregar a lista da tabela, e por fim duas caixas de seleção que permitem filtrar a exibição tanto das observações quanto das consultas. Para salvar/carregar a lista, caso um arquivo não tenha sido anteriormente indicado, aparecerá uma caixa de diálogo para a seleção do arquivo. Para incluir manualmente uma variável à lista, o nome da variável (conceitos ou papéis instanciados para indivíduos específicos do domínio, ou seja,  $C(1)$ ,  $r(0,1)$ , etc) deve ser adicionado pelo campo **Set Variable**; a variável aparecerá na lista porém não estará atribuído a ela nenhum valor. Para atribuir um valor à variável (ou alterar o valor da mesma) deve-se clicar na linha da



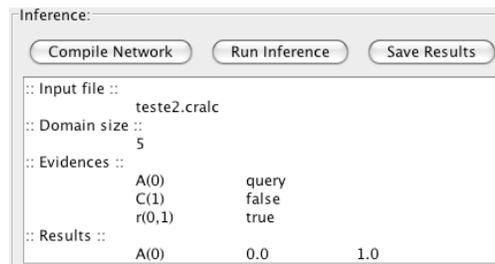
Figura 3: Especificação da terminologia.

tabela correspondente a ela e selecionar um valor dentre as opções que aparecerão: **query** a estabelece como variável de consulta, **true** observa a variável como *verdadeiro*, **false** observa a variável como *falso* e **remove** exclui a variável da lista. Na Figura 4(a) pode-se ver um exemplo onde está sendo associado o valor **true** para a variável  $C(1)$ .

A região inferior contém os botões a serem utilizados para executar a inferência e dispõe de um campo para a exibição dos resultados. O botão **Compile Network** gera as fatias do modelo de acordo com as observações e consultas, exportando-as automaticamente como redes Bayesianas em arquivos `.bif`, enquanto que o botão **Run Inference** executa a inferência e exhibe os resultados, como pode ser visto na Figura 4(b). Independentemente dos parâmetros utilizados na chamada do programa, no modo com interface gráfica os resultados não são automaticamente salvos (diferentemente do modo para linha de comando). Para salvar os resultados, é necessário clicar no botão **Save Results**, que os escreverá num arquivo com o nome fornecido para os resultados (parâmetro `[-r name]`). Na ausência de um nome, será utilizado o nome *Untitled*; para fácil referência, o nome dos resultados aparece no título da janela da interface.



(a) Inclusão de evidência e consulta.



(b) Exibição dos resultados.

Figura 4: Consulta, evidência e resultado.

Por fim, a janela da interface gráfica ainda apresenta um campo de status na borda inferior da janela, onde é exibido feedback das atividades em andamento. A janela conta também com três menus na borda superior da janela. O primeiro menu, **File Name**, contém duas opções: **Results** permite definir e alterar o nome do arquivo onde serão salvos os resultados (assim como o nome dos arquivos com as fatias e o título da janela), enquanto que **Observations** permite definir e alterar o nome do arquivo com a lista de observações e consultas. O segundo menu, **Export**, contém apenas uma opção, para proposicionalizar a terminologia carregada e salva-la como uma rede Bayesiana. E finalmente o terceiro menu, **Settings**, apresenta opções de configuração do programa: **Verbose** ativa a exibição de informações adicionais sobre o andamento das tarefas no campo de resultados da janela; **File type** permite escolher o formato a ser utilizado para exportar as fatias e para exportar a rede proposicionalizada; e **Inference** permite escolher o método de inferência a ser utilizado dentro das fatias.

**Arquivos de entrada:** A linguagem escolhida para representar uma terminologia em *CRALC* segue a *Description-Logic Knowledge Representation System Specification* (KRSS). A especificação completa da KRSS encontra-se disponível em <http://dl.kr.org/krss-spec.ps> e embora seja uma especificação completa para lógicas de descrição, a *CRALC* utiliza apenas de um pequeno subconjunto da especificação, listado abaixo.

(and $C_1 \dots C_n$ )	$C_1 \sqcap \dots \sqcap C_n$
(or $C_1 \dots C_n$ )	$C_1 \sqcup \dots \sqcup C_n$
(not $C$ )	$\neg C$
(all $r C$ )	$\forall r.C$
(some $r C$ )	$\exists r.C$
(define-concept $C D$ )	$C \equiv D$
(define-primitive-concept $C D$ )	$C \sqsubseteq D$

Note que em (define-concept  $C D$ ) e em (define-primitive-concept  $C D$ ),  $D$  pode ser substituído por qualquer expressão tratável pela *CRALC*, com uma exceção: quantificadores devem ser definidos separadamente. Por exemplo,  $A \sqsubseteq B \sqcup \forall r.C$  deve ser escrito como  $A \sqsubseteq B \sqcup D$  com  $D \equiv \forall r.C$

Inclusões de probabilidade são representadas por:

(probability $B \alpha$ )	$P(B) = \alpha$
(conditional-probability $B A \alpha$ )	$P(B A) = \alpha$

e assim como no caso anterior, onde temos (conditional-probability  $B A \alpha$ ),  $A$  pode ser substituído por qualquer expressão tratável pela *CRALC*, salvo a exceção já descrita.

Quebras de linha, espaços e tabulações podem ser utilizados para formatar o arquivo e comentários podem ser adicionados caracterizados por ‘;’. Abaixo pode-se ver um exemplo de um arquivo válido.

---

```
;; This is a valid input file
(probability A(x) 0.7)
(probability B(x) 0.4)
(define-concept C(x)
  (and
    A(x)          ;; note that comments can be
    (not B(x) )   ;; used in the middle of sentences.
  )
)
```

---

Os arquivos contendo as variáveis da consulta e as observações são ainda mais simples. Cada linha deve conter o nome da variável seguido do valor atribuído ao mesmo (`query`, `true` e `false`). Comentários são permitidos, no mesmo formato já descrito.

---

```
;; This is a valid file
A(1)   query
B(0)   true
r(0,1) false
```

---

## Referências

- [1] Fabio G. Cozman and Rodrigo B. Polastro. Loopy propagation in a probabilistic description logic. In Sergio Greco and Thomas Lukasiewicz, editors, *Second International Conference on Scalable Uncertainty Management*, Lecture Notes in Artificial Intelligence (LNAI 5291), pages 120–133. Springer, 2008.