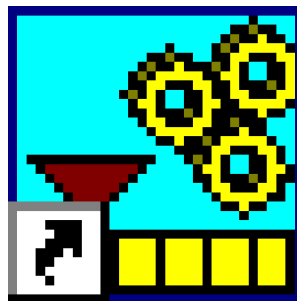


V TURING



MANUAL



2007

*Paulo Jorge Alves
Dos santos*

Índice

Manual do Visual Turing

Nota Prévia

Razões e Particularidades deste Manual	pág. 04
--	---------

Introdução

Uma breve Introdução às Máquinas de Turing	pág. 06
Uma breve Introdução ao Visual Turing	pág. 08

Objectos de Programação em Visual Turing

Instruções	pág. 10
Setas e Condições	pág. 11

Desenvolvimento das Máquinas de Turing

Editor Gráfico

Introdução ao Editor Gráfico	pág. 14
Criar um novo Projecto	pág. 15
Inserir uma nova Máquina	pág. 16
Inserir um novo Símbolo	pág. 17
Inserir uma nova Variável	pág. 18
Inserir uma nova Instrução	pág. 19
Inserir um ramo (uma ramificação)	pág. 20
Inserir uma Seta	pág. 21
Modificar a Condição da Seta	pág. 22
Seleccionar objectos	pág. 23
Apagar objectos	pág. 24
Cortar/Copiar/Colar (Cut/Copy/Paste)	pág. 25
Fazer anotações na Máquina	pág. 26

Editor da Fita

A Fita da Máquina em Visual Turing	pág. 28
Modificar a Fita	pág. 29
Guardar (saving) a Fita	pág. 30
Correr (running) e Depurar (debugging) a Máquina	
Correr (running) um Programa	pág. 32
Depurar (debugging) um “Programa-Gráfico”	pág. 34

Exemplos de Máquinas

Problemas de Teoria da Computação	pág. 36
Biblioteca de Exemplos	pág. 37

Fontes de Informação

Fontes electrónicas e URLs.....	pág. 47
Fontes bibliográficas	pág. 48

Nota Prévia

Razões e Particularidades deste Manual

Porquê este Manual ?

O presente **Manual do Visual Turing** foi concebido, antes de mais, para cursos de formação, com funções de suporte aos trabalhos na cadeira de Inteligência Artificial, i ou sistemas de computação.

O Manual baseia-se nos *Contents* e no serviço de *Help* do Visual Turing, versão 1.0, do Informático romeno Cristian Cheran (cheran@cheran.ro) (<http://www.cheran.ro/vturing>). Começou por ser uma mera tradução, e evoluiu para um trabalho de concepção e realização autónomo.

É muito possível que o presente Manual acabe por abrir caminho a conhecimentos mais profundos por parte dos alunos, é dirigida aos estudantes e estudiosos de cadeiras como *Teoria da Computação* ou, mesmo, de *Inteligência Artificial*.

Que o Manual vos seja de algum modo útil. E com a dedicatória, vão os votos de bom trabalho.

Covilhã,

Sábado, 17 de Fevereiro de 2007

Paulo Jorge Alves Dos Santos

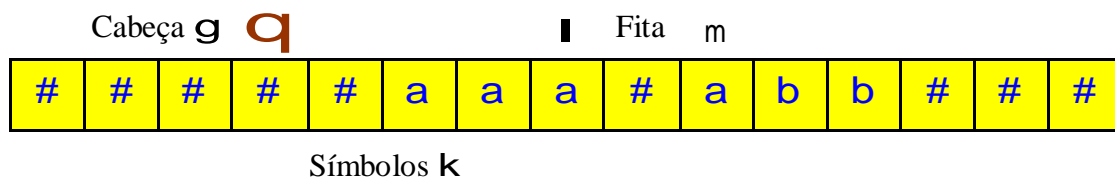
Introdução

Uma Breve Introdução às Máquinas de Turing

O que é uma Máquina de Turing?

Uma **Máquina de Turing** é uma máquina abstracta, lógico-matemática. Consiste numa **fita** de comprimento conceptualmente infinito e dividida em “**células**”. A fita pode estar vazia ou preenchida com **símbolos**, sendo que cada símbolo só pode estar inserido numa e só numa dada célula particular. (No Visual Turing, como teremos oportunidade de ver mais adiante, os **símbolos definidos pelo utilizador** são letras e cada célula vazia é, **convencionalmente**, representada pelo símbolo “#”.) A fita tem ainda uma “**cabeça**” que pode

- (1) *ler o conteúdo de cada célula,*
- (2) *inserir símbolos,*
- (3) *deslocar-se, de cada vez, uma célula para a esquerda ou uma célula para a direita.*



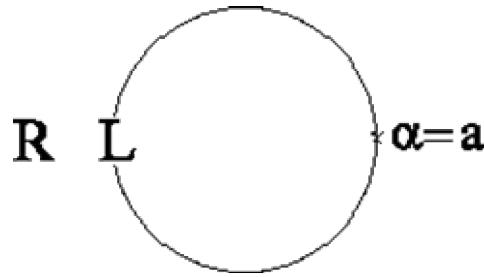
A maneira como a cabeça se desloca ou escreve símbolos é determinada pelo “**programa**”. Em Visual Turing (VT), o programa é um “**programa-gráfico**”, ou seja, um grafo cujos nodos são constituídos por instruções e cujos arcos estabelecem a ligação entre instruções.

Há instruções especiais, como **L** (desloca a cabeça da fita uma célula para a esquerda), ou **R** (desloca a cabeça da fita uma célula para a direita), a “**instrução símbolo**” – que escreve um dado símbolo na célula; e a “**instrução variável**” – que escreve o valor assumido pela variável em causa na célula.

As linhas (aliás, arcos) ligam os nodos do grafo e determinam, assim, o caminho de execução. Os arcos estão associados a certas condições, quer dizer, se as condições se verificarem o caminho de execução segue para diante nessa linha (arco) até ao nodo de destino. (Uma condição é verificada - isto é, “é verdadeira” - se o símbolo alocado abaixo da posição da cabeça se encontra em qualquer dos operandos à direita da condição.)

Um dos elementos mais importantes são as variáveis. As variáveis têm o mesmo significado e cumprem o mesmo papel que nas linguagens de programação – ou seja, contêm valores. As variáveis podem escrever-se na fita ou podem introduzir-se (*carregar-se*) da seguinte maneira: quando uma variável é mencionada numa condição como o operando da esquerda – induzindo a ideia de atribuição -, se a condição se revelar verdadeira, num dado momento da execução, a variável assume o valor da posição da cabeça (o símbolo lido pela cabeça).

As variáveis são representadas por símbolos (letras) gregos.



Um programa simples que se desloca uma vez para a direita e depois para a esquerda se e enquanto o símbolo lido for **a** . A variável α (alfa) é carregada com o símbolo lido.

Na expressão das condições pode usar-se a negação, quer dizer, se o símbolo lido for diferente dos símbolos inscritos na condição, a execução prossegue adiante, por aí; e carrega a variável se houver uma como operando à esquerda.

Podemos usar máquinas como instruções. Por exemplo, a clássica **L#** que “varre” a fita à esquerda até encontrar a primeira célula vazia (portanto, convencionalmente, com o símbolo #). Nestes casos, as máquinas (os respectivos ícones representativos) são inseridas no programa gráfico como quaisquer outras instruções, e são executadas do mesmo modo – algo semelhantes a um procedimento sobre funções em C ou Pascal.

Há uma diversidade enorme de computações executáveis com uma Máquina de Turing: adição, subtração, multiplicação ou operações com “strings”, como copiar, “saltar”, ... Muitos destas computações podem encontrar-se no Visual Turing, como exemplos. No presente Manual, incluímos três exemplos da “biblioteca” freeware que o Visual Turing traz geralmente como ilustrações das capacidades do VT.

Uma Breve Introdução ao Visual Turing

O que é o Visual Turing ?

Embora numa abordagem muito geral e sumária, vimos até agora o que é uma Máquina de Turing. E o Visual Turing, o que é? Fique desde já marcada aqui que se tratam de conceitos e “objectos” correlacionados mas distintos. Um (a Máquina de Turing, concebido por Alan Turing) é um conceito abstracto e universal de Máquina cujas propriedades traduzem as propriedades gerais da Computação, outro (o VT, de Cristian Cheran) é uma ferramenta que nos permite implementar e visualizar Máquinas de Turing, ou seja, computações.

O **Visual Turing** é tecnicamente um IDE (Integrated Development Environment), isto é, um ambiente de desenvolvimento completamente integrado que permite expressar conceitos e problemas da computação segundo as Máquinas de Turing.

O **Visual Turing** suporta todo o processo de desenvolvimento de Máquinas de Turing. Suporta o desenho do grafo, que é, na prática, o “programa” da Máquina e se apresenta sob uma forma gráfica, visual, muito *user friendly*, ou seja, muito simpática ao utilizador. Permite editar a fita da Máquina e permite correr o Programa.

O **Visual Turing** é extremamente pedagógico. Não apenas a sua apresentação gráfica é apelativa, mas múltiplos detalhes e funcionalidades denotam esta preocupação. Por exemplo: o Programa pode correr em diversas velocidades, possibilitando um exame minucioso da sua execução.

*Objectos
de Programação em
Visual Turing*

Instruções em Visual Turing

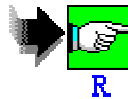
As Instruções são representadas por pequenos ícones inscritos num quadrado e sobre um fundo de cor (geralmente verde, mas também azul turquesa ou amarelo). As instruções clássicas são:

- a) **L** para movimentar a cabeça da Máquina uma casa (uma célula) para a esquerda;
- b) **R** para movimentar a cabeça da Máquina uma casa (uma célula) para a direita;
- c) a **instrução símbolo**, para escrever um símbolo na fita;
- d) a **instrução variável**, para escrever o valor da **variável** na fita;
- e) a **instrução máquina**, para chamar outra **máquina** e executar o seu programa.
- f) a instrução **nothing**, introduzida por **Cristian Cheran** no VT; trata-se de uma instrução suplementar, que não faz nada (como o seu nome indicia), mas que, com este artifício, permite a combinação de duas setas e obter assim uma condição conjuntiva (AND).



As Instruções em Visual Turing

As instruções podem ser combinadas nos “programas gráficos” das máquinas. Todas as máquinas têm uma **instrução inicial** – a instrução que é executada em primeiro lugar. A instrução inicial apresenta-se sempre com uma seta preta à esquerda do quadrado da instrução respectiva, como na figura abaixo:



Uma instrução inicial

Para facilitar o processo de depuração (*debugging*), torna-se conveniente ter “pontos de paragem” (*breakpoints*) nalgumas instruções. Para esse procedimento, recorra ao menu de contexto ou à caixa de diálogo das propriedades. Uma instrução com marca de “ponto de paragem” apresenta-se como a instrução abaixo, ou seja, com uma etiqueta vermelha e a indicação “stop”, no canto superior esquerdo. Tenha presente que os “pontos de paragem” interrompem o programa apenas na modalidade de depuração visual (*visual debugging mode*).



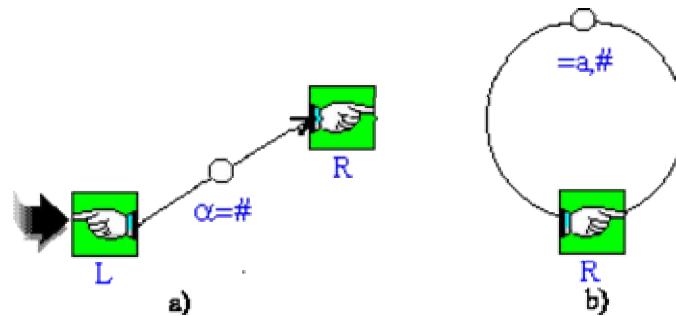
Uma instrução com “ponto de paragem” (breakpoint)

Setas e Condições em Visual Turing

As setas no Visual Turing ligam duas instruções. Para facilitar a edição, há dois tipos de setas no VT:

- a) as setas normais, que ligam duas instruções diferentes no grafo; e
- b) as “setas auto-referentes” (self arrows) que ligam uma instrução a si própria.

A distinção justifica-se pela forma diferente como as setas se representam.

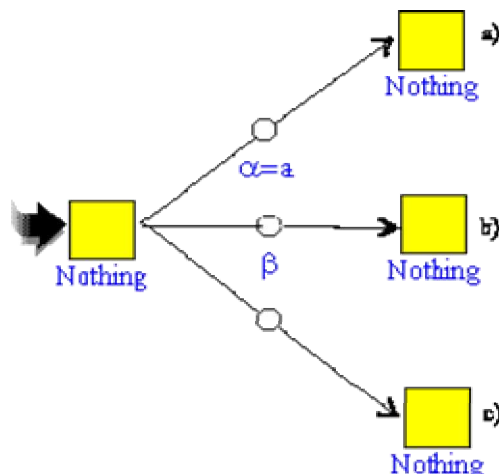


Os dois tipos de setas em Visual Turing

O papel assumido pelas setas é o de caminhos de execução. Num dado momento da execução, o programa segue um ou outro caminho consoante a seta que apresentar a condição verdadeira (uma condição é verdadeira se o símbolo lido pela cabeça da máquina na fita é igual a um dos símbolos ou variáveis do lado direito da condição). Se esse caminho é escolhido, a variável no lado esquerdo da condição – a existir – é carregada com o valor lido pela cabeça, e a instrução indicada pela seta torna-se a instrução a executar.

A ordem de prioridades das condições obedece a uma classificação de importância relativa das mesmas. Há

- a) **condições normais**, com alguns símbolos mencionados no lado direito, com a prioridade máxima;
- b) **condições de carregamento de variáveis**, que carregam as variáveis com o símbolo lido pela cabeça; e
- c) **condições disjuntivas** (*else conditions*), com prioridade mínima.



Prioridades na avaliação dos caminhos possíveis

*O Desenvolvimento
das
Máquinas de Turing*

*O Desenvolvimento
das
Máquinas de Turing*

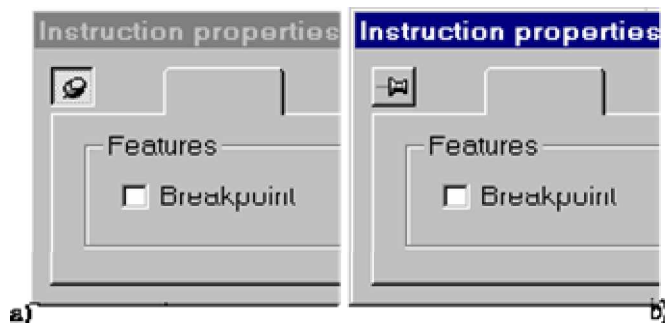
- O Editor GRÁFICO

Introdução ao Editor Gráfico

O Visual Turing tem um poderoso Editor para o "programa-gráfico". O Editor suporta vários níveis de operações UNDO (ANULAR OPERAÇÕES), bem como as clássicas operações CUT-COPY-PASTE (CORTAR/COPIAR/COLAR).

Qualquer operação que faça no editor – mover ou inserir instruções ou setas – tem um retorno visual. É possível seleccionar múltiplos objectos no editor para os deslocar ou fazer passar para o *clipboard*. Qualquer instrução, seta ou máquina, etc., tem um menú de contexto e uma caixa de diálogo que permite uma edição rápida dos atributos de um dado objecto.

Para obter o menú de contexto basta clicar no botão direito do rato sobre o objecto em causa. Para obter as propriedades, deve seleccionar a opção **Propriedades (Properties)** do menú de contexto ou duplo-clicar o objecto se se tratar de uma instrução ou seta. Quando se recorrer à caixa de diálogo das **Propriedades**, pode a) afixar-se com um prego a caixa na “frame” (para consulta e verificação sistemática, prolongada das propriedades), ou b) pode deixar-se a caixa livre. Quando a caixa está fixa, o facto de não estarmos focados nela (de a janela não estar activa), não a fecha; mantem-se assim o efeito imediato sobre o objecto editado e seleccionado.



Todas as operações para editar o gráfico estão disponíveis na barra de ferramentas principal. Pode ver-se o que faz cada botão ao “sobrevoar” com o rato por cima de cada um deles. Uma explicação mais pormenorizada do comando pode encontrar-se na barra de estado.



Para maior comodidade e facilidade no processo de edição, pode introduzir-se uma grelha (*grid*). Para isso, accione a opção de menú **View/Grid**.

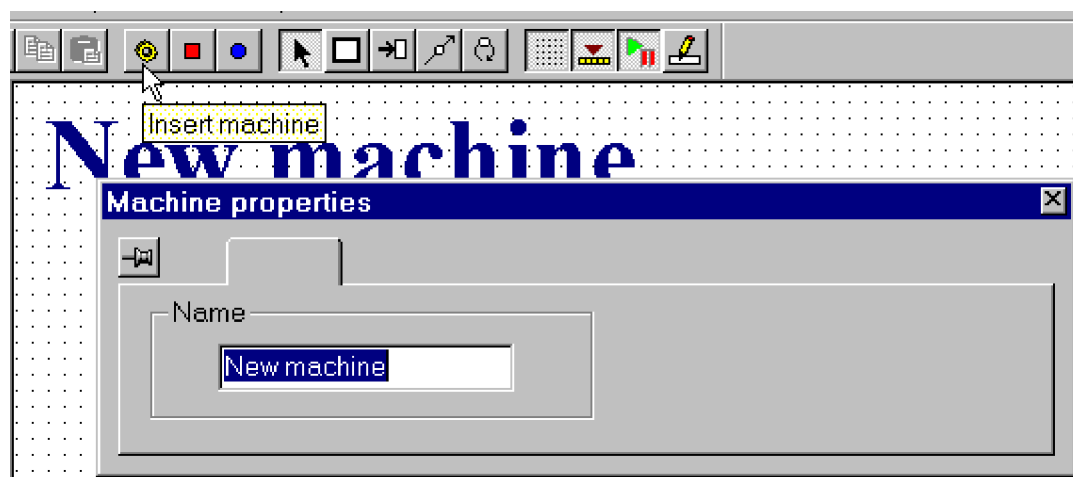
Criar um novo Projecto

Quando se abre o Visual Turing, abre-se imediatamente um Projecto, por defeito. Pode também dar-se início a um novo Projecto recorrendo à opção de menú **File/New** ou, ainda, pressionando o botão **New** na barra de ferramentas principal.

O projecto que se abre por defeito tem uma única máquina designada por “Main” (ou seja, “Máquina Principal”). Esta designação pode ser alterada, bem como o símbolo usado para todas as Máquinas de Turing: #, o símbolo da célula em branco. Para essa alteração, basta que recorra à caixa de diálogo das propriedades deste procedimento. Note, porém, que durante o período de desenvolvimento não consegue apagar o símbolo nem a máquina principal inserida por defeito. Para prosseguir com o seu trabalho (Projecto) deve começar por colocar lá o seu gráfico, isto é, o seu Programa-Gráfico.

Inserir uma nova Máquina

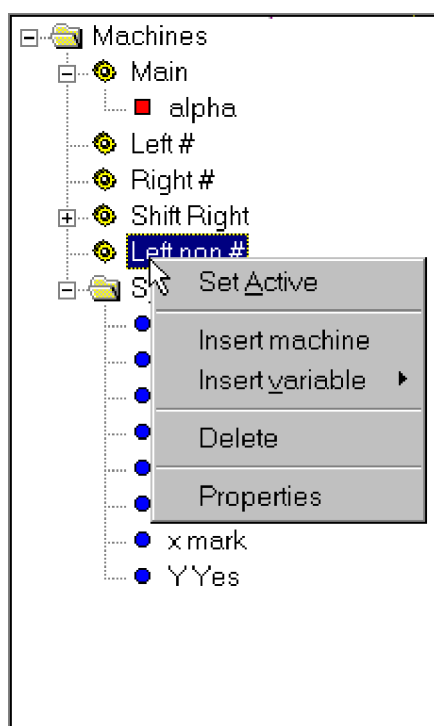
Na Programação de Turing é permitido o uso composto de máquinas, isto é, o uso de máquinas como instruções no programa gráfico, algo parecido com procedimentos. O Visual Turing suporta este tipo de construções compostas. Para a inserção de um procedimento deste tipo compósito, deve seleccionar a opção de menú **Tools/Insert Machine** ou um botão da barra de ferramentas principal – cf. a figura abaixo.



O cursor está sobre o botão para inserir uma nova Máquina.

Veja a caixa de diálogo das propriedades da Máquina

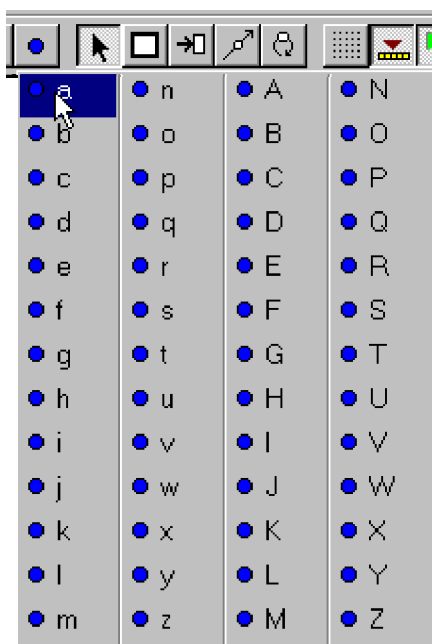
As máquinas podem ser seleccionadas a partir de um painel especial – em formato de árvore, ou cascata – o qual mostra todos os procedimentos e variáveis relacionadas – cf figura abaixo.



Veja o menu de contexto para a Máquina.

Inserir um novo Símbolo

Os símbolos podem ser inseridos recorrendo à opção de menú **Tools/Insert symbol** ou ao botão correspondente na barra de ferramentas principal – cf. figura abaixo.

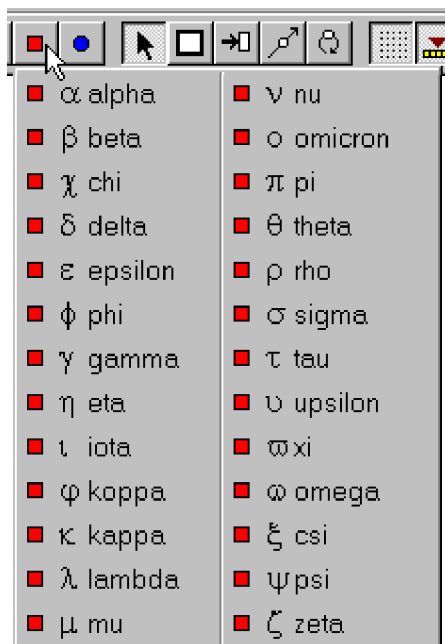


Os símbolos inserem-se premindo o botão com um pequeno círculo azul

Os símbolos podem escrever-se na / ou serem lidos a partir da fita e, de certa maneira, representam uma espécie de “moeda” na Máquina de Turing. Em VT, os símbolos que podem ser usados vão de “a” a “z”, e de “A” a “Z”. O VT inovou num pequeno pormenor: é possível atribuir “nicknames” (um misto de “diminutivo”, “alcunha” e “nome carinhoso”) aos símbolos. Recorrendo à caixa de diálogo das propriedades, pode atribuir-se um “nickname” a um símbolo que aparece ao fundo da respectiva instrução-símbolo.

Inserir uma nova Variável

Para inserir uma nova variável, deve recorrer-se à opção do menu **Tools/Insert variable** ou ao botão na barra de ferramentas principal, cf. a figura abaixo:



Os pequenos quadrados vermelhos representam variáveis

Os Programas em Visual Turing usam símbolos Gregos para as variáveis. Para maior comodidade, no Visual Turing os nomes dos símbolos Gregos acompanham os respectivos símbolos; por exemplo, o nome *kappa* segue-se ao símbolo *k*. As variáveis – elas próprias símbolos – usam-se para representar símbolos. O carregamento (*loading*) de uma variável é feito colocando-a como o operando esquerdo de uma condição numa seta – cf. *Modificar a condição da seta*.

(Na versão a que nos estamos a reportar, a lista dos nomes e símbolos gregos contém dois erros, algo grosseiros:

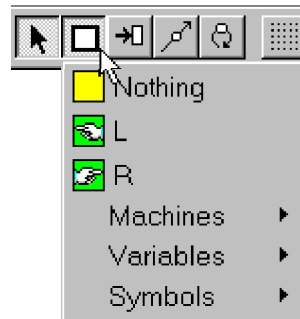
- (1) um (o menor, apesar de tudo), consiste no facto de a lista não se encontrar ordenada segundo critérios alfabéticos convencionais;
- (2) o segundo, porque inclui letras ou símbolos que não são convencionalmente aceites e/ou com erros de denominação.

O assunto merecerá uma correcção directa na codificação na respectiva opção de menú.)

Inserir uma nova Instrução

Na programação de Turing o elemento essencial é a instrução.

Pode inserir uma nova instrução recorrendo à opção de menu **Tools/Insert instruction** ou ao botão da barra de ferramentas principal – cf. figura abaixo.

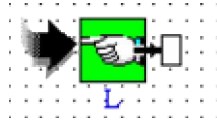


Inserir uma nova instrução com a barra de ferramentas

Ao pressionar o botão esquerdo do rato, a instrução aparece e pode então arrastá-la para onde quiser. Quando descomprime o botão, a instrução estará efectivamente inserida no gráfico.

Inserir um novo Ramo

Uma característica muito útil do editor é a ferramenta de ramificação. Ao seleccionar **Tools/Insert branch** ou o botão na barra de ferramentas, podemos estender o programa gráfico adicionando um ramo. O que é um ramo? É tão simplesmente uma seta que liga a instrução seleccionada a outra. A inserção de uma seta e da nova instrução são feitas num só passo indissociável. Para isso, escolha a instrução “de destino” e “sobrevoe” sobre a instrução de partida. A forma do cursor dá uma indicação da direcção da ramificação – se a houver – como na figura abaixo. Quando se clica, o ramo aparece.

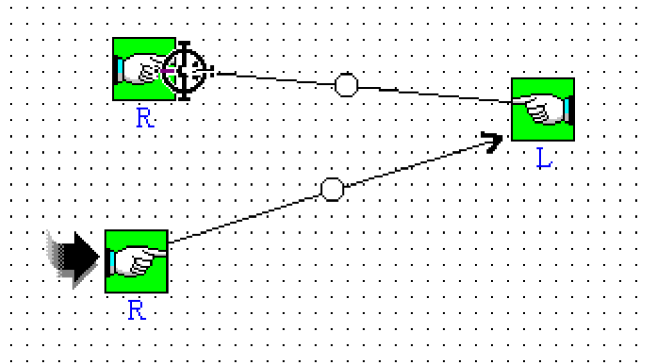


Veja a forma do cursor

Inserir uma nova Seta

Recorrendo à opção de menú **Tools/Insert arrow** ou **Tools/Insert self arrow** ou, ainda, aos botões respectivos da barra de ferramentas principal podem inserir-se setas para estabelecer a ligação com instruções – os nodos do grado. Veja [Setas e condições](#) para informações mais detalhadas.

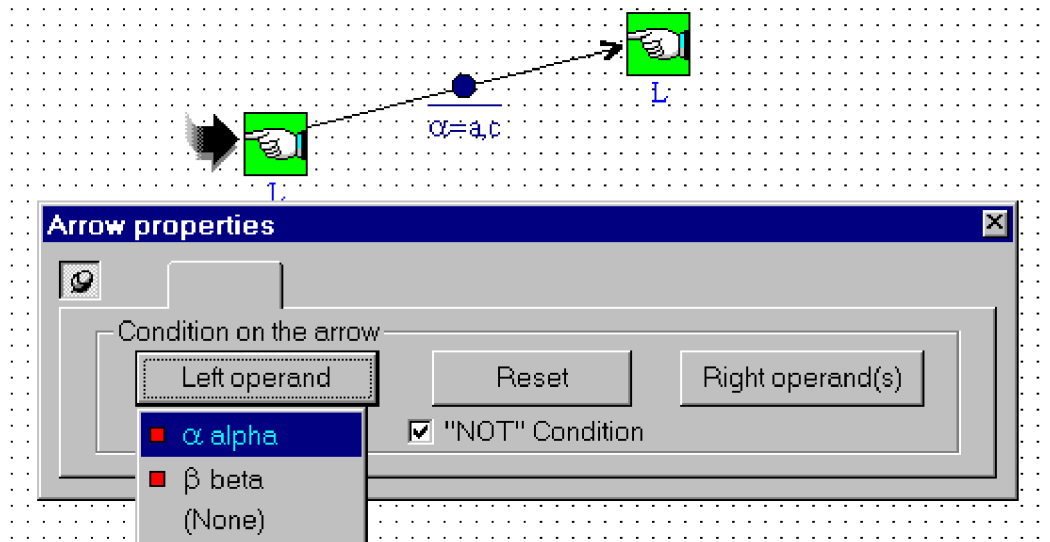
Depois da seleccionar um comando de inserção de seta, pode sobrevoar sobre as instruções para seleccionar a instrução de partida. Quando o cursor se transformar num alvo pode clicar o botão esquerdo do rato e, com isso, seleccionar a instrução de partida. Pode então arrastar a seta para outra posição para a seta de auto-referência ou para a segunda instrução para as setas normais. Quando clicar de novo, a inserção estará consumada.



Veja a forma que o cursor assume quando a seta se liga à instrução de destino

Modificar a Condição da Seta

Depois de ter inserido uma seta a ligar duas instruções, pode mudar a sua condição – veja [Arrows and conditions](#) para mais informação. A partir do menú de contexto da seta, seleccione **Propriedades (Properties)**. Obterá a caixa de diálogo das propriedades da seta, a partir da qual poderá modificar as condições com os menús.



A caixa de diálogo das Propriedades das Setas (*Arrow properties*)

Apagar (*deleting*) objectos

Pode apagar objectos, primeiro seleccionando-os e de seguida usar a opção de menú **Edit/Delete** ou o botão na barra de ferramentas principal. Todas as operações de edição – ou seja, apagar (*delete*), cortar/copiar/colar (*cut/copy/paste*) -, encontram-se também disponíveis no menú de contexto da máquina corrente, menú que podemos obter com um simples clique no fundo do écran da máquina.

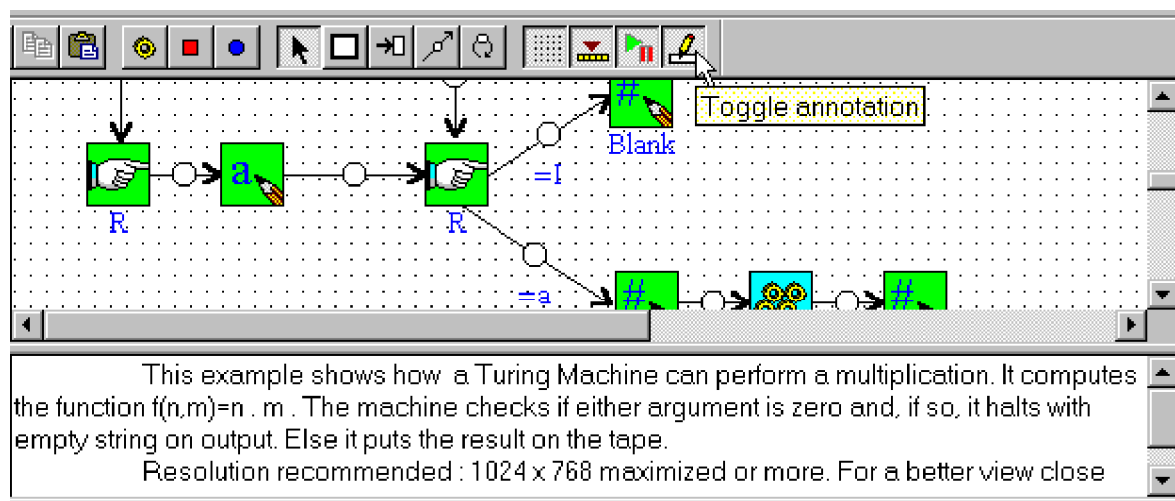
Deve ter-se em atenção que quando se apaga uma ou mais instruções, todo o conjunto de setas que lhe estão associadas também sofrem desta operação, ou seja, são também apagadas.

Cortar/Copiar/Colar (*Cut/Copy/Paste*)

Em Visual Turing o recurso ao clipboard torna-se extremamente eficiente em tudo o que sejam as operações de edição “cortar”, “copiar” ou “colar”. Deve ter-se em atenção que, seleccionada uma ou mais instruções, estas podem facilmente ser depositadas no *clipboard*; mas apenas as setas que lhe estão associadas (a essa ou essas instruções) farão parte desse depósito no *clipboard*. Ou seja, apenas as setas que tenham ambas as instruções de origem e de destino incluídas na selecção feita. Após este depósito no clipboard, os conteúdos depositados podem, em qualquer momento, ser “colados”.

Anotar uma Máquina

Uma facilidade disponibilizada pelo VT, aparentemente menos interessante mas que se pode revelar de grande pertinência, é a possibilidade de fazer anotações sobre as máquinas. Podemos proceder à anotação e, em qualquer momento, editar os conteúdos das observações anotadas. Quer na barra de ferramentas principal, ou simplesmente usando o dispositivo corrente do tipo “split” com o cursos, há uma janela com um campo de dição onde poderão proceder-se às anotações (veja a partição da janela, em baixo, com as anotações).



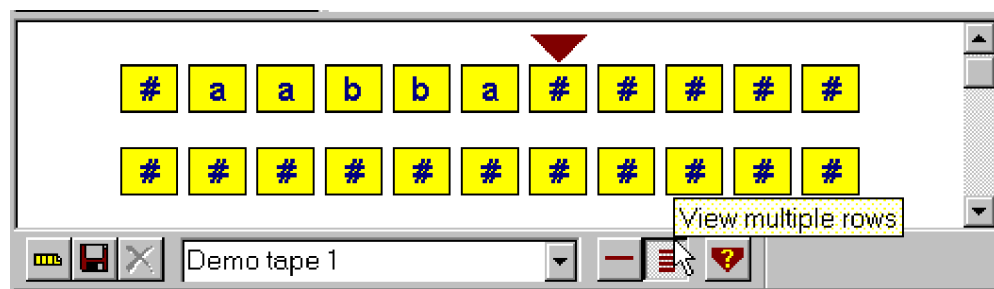
Use o botão por baixo do cursor para fazer anotações

*O Desenvolvimento
das
Máquinas de Turing*

- O Editor da FITA

A Fita no Visual Turing

Há um painel especial para a visualização e edição da fita, como pode ver pela figura abaixo.

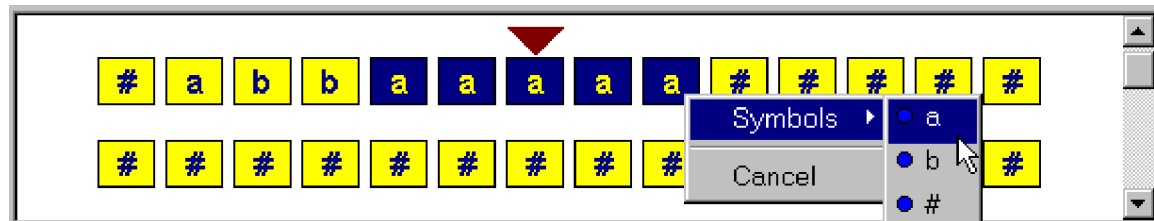


O painel da Fita

Em VT pode ver-se a fita como uma única linha – a visualização clássica -, ou como várias linhas. No exemplo acima a fita apresenta-se sob a forma de múltiplas linhas. Para optar por uma ou outra forma de apresentação, basta premir um ou outro dos dois botões que se vêem na barra de ferramentas da fita.

Modificar a Fita

Os conteúdos da fita podem ser editados no painel da fita. Com recurso ao rato e a selecções múltiplas das células da fita, podemos mudar estes mesmos conteúdos através do menú de contexto – cf. figura abaixo.

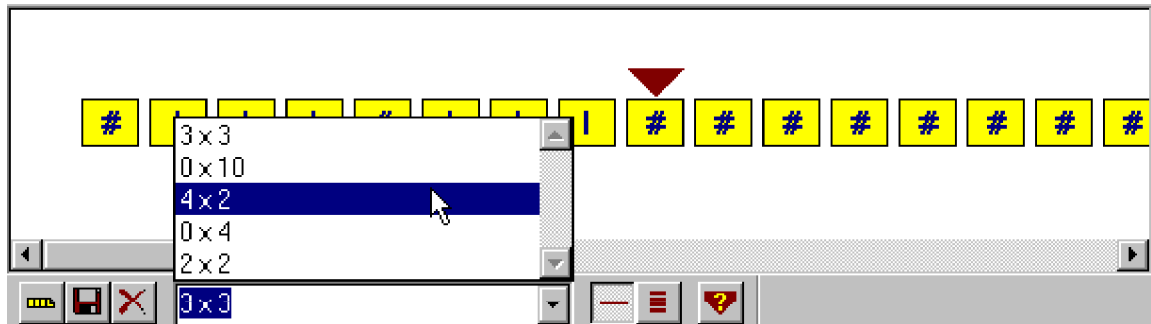


Use o menu de contexto para modificar o conteúdo das células da fita

Para editar a fita podemos usar o teclado do nosso computador. Depois de seleccionarmos uma ou mais células da fita, pode recorrer às teclas com setas para se movimentar de uma célula para a outra. Se mantiver premida a tecla SHIFT pode fazer múltiplas selecções. Para apagar os conteúdos de uma selecção, use a tecla DEL. Para mover a cabeça para a célula seleccionada, pressione a tecla ENTER.

Guardar (*saving*) a Fita

Em Visual Turing podemos guardar as configurações das fitas que quisermos, para mais tarde carregarmos de novo e fazermos correr o programa sobre essa configuração. Há, para isso, um conjunto de três botões muito úteis na barra de ferramentas: um que permite a criação de novas configurações de fita com nomes que podemos escolher, outro que permite guardar a fita em uso corrente e outro que permite apagar fitas. As fitas podem ser carregadas quando seleccionamos uma das configurações e “largamos” o botão de selecção, ficando essa configuração visível no campo de edição junto aos três botões referidos. Veja-se o exemplo abaixo, relativo à fita com as configurações gravadas sob os nomes 3x3, 4x2 e outras: a configuração gravada sob o nome 3x3 é a primeira na lista e é aquela que, por esse facto, por defeito, se apresenta visível no campo de edição; a configuração 4x2 é a gravação que queremos seleccionar, por isso passará a seguir a visualizar-se no campo de edição mal se “largue” o cursor que a seleccionou – como se vê na figura –; os restantes nomes gravados (0x10, 0x4, etc) correspondem a configurações gravadas que ficam “em arquivo” para uso quando necessitarmos. As fitas estão devidamente gravadas no ficheiro de Projecto.



Seleccionar uma Fita gravada

*O Desenvolvimento
das
Máquinas de Turing*

- Correr e depurar a Máquina

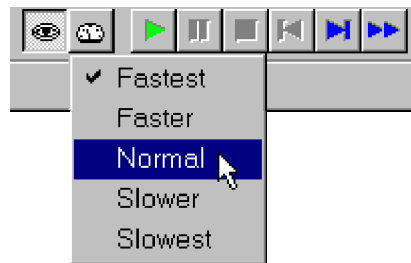
Correr um Programa

Depois de termos “desenhado” o programa gráfico, podemos de imediato executá-lo. Para isso, podemos recorrer à barra de ferramentas principal e seleccionar a opção **Play** do menu **Run**. Mas, a forma mais rápida e intuitiva é recorrendo à barra de play (**play**, “tocar”) do Visual Turing (cf. figura abaixo):



A barra de “play”

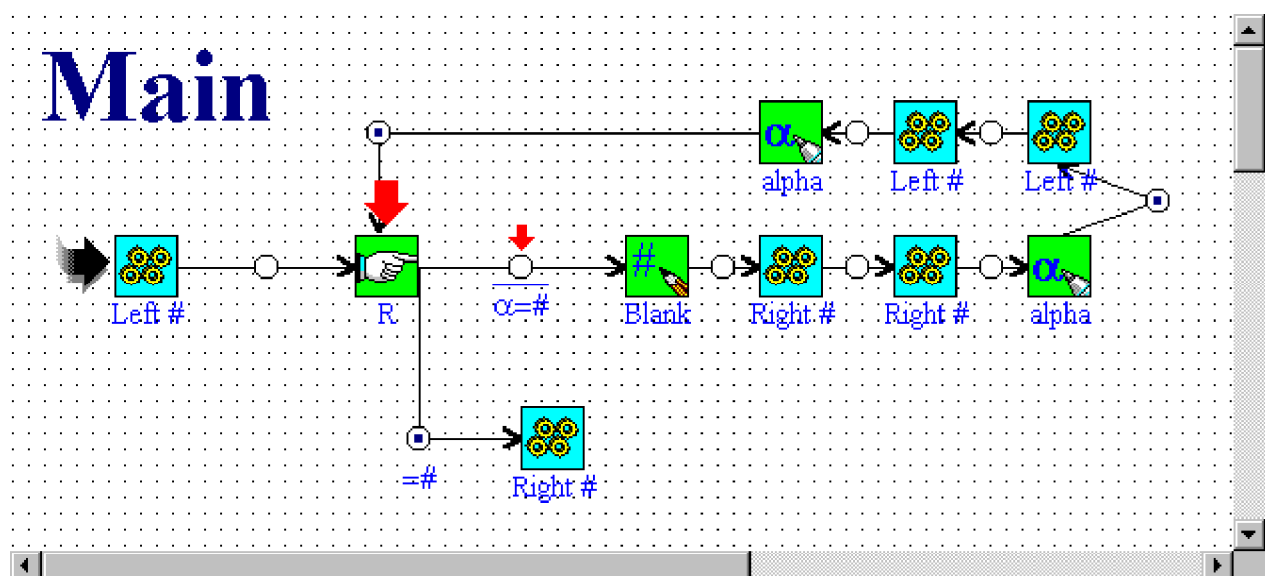
Os botões são semelhantes aos de um gravador de vídeo, portanto muito simples e sugestivos, de fácil utilização. Quando uma máquina está a correr, há opções que facilitam a sua correcção (depuração, *debugging*). A primeira e mais importante é a possibilidade dinâmica de alteração da velocidade de execução da máquina. Há 5 velocidades disponíveis.



O menu da velocidade

Para analisar e depurar (corrigir) um programa, há mesmo uma modalidade especial que pode ser accionada pressionando o botão do “olho” na barra de *play* – o modo de depuração visual (**visual debugging mode**). Quando a máquina se encontra neste modo de depuração visual, todos os passos do programa são visíveis no écran e encontram-se disponíveis as opções especiais de depuração – a saber, os pontos de paragem (*breakpoints*) e a retro execução (*reverse execution*).

Quando estamos a observar uma máquina a correr (em execução), podemos ver o traçado da execução – cf. figura abaixo.

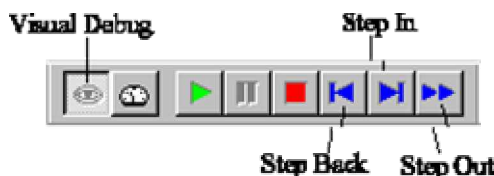


Correr uma Máquina

Uma grande seta vermelha aponta para a instrução corrente no traçado de execução e uma pequena seta (também vermelha) mostra o caminho para a instrução seguinte. Quando não aparece a pequena seta vermelha é porque a máquina em questão está em fase terminal de execução. Significa isso que, se se trata de uma submáquina, essa submáquina está liminarmente terminada, e a execução continuará “saltando” de novo para o ponto (isto é, para a instrução) que “chamou” a referida submáquina. Se se trata da máquina principal, então a execução está mesmo totalmente terminada.

Depurar (*debugging*) um Programa

Depois de ter desenvolvido o programa gráfico, é possível depurar (*debugging*) as estruturas de “código” recorrendo a vários dispositivos de depuração disponíveis em VT. Sempre que se queira depurar uma máquina, recomenda-se que se entre na modalidade de depuração visual (**visual debug mode**). Deve ter-se sempre presente que, ao entrarmos em modo de depuração visual, passamos a ver todo e cada um dos movimentos de execução do programa; nada melhor, portanto, para uma análise e monitorização daquilo que se concebeu e programou!. Veja-se a figura abaixo e atente-se nos utilíssimos botões para depuração.



As opções para Depurar (*debugging*)

Quando estamos a depurar um programa, podemos e devemos usar pontos de paragem (*breakpoints*) nas instruções. Pode apor-se um ponto de paragem numa instrução usando o menu de contexto dessa instrução. O ponto de paragem (*breakpoint*) introduz uma pausa na execução na posição imediatamente antes da instrução ser executada. Note-se que esta opção está disponível apenas na modalidade de depuração visual (*visual debug mode*).

Opção Step Into. Com esta opção (**Step Into**, isto é, metaforicamente, *vamos pra diante!*) faz-se executar um passo adiante. Esta funcionalidade está disponível tanto em modo de depuração visual como em modo normal.

Opção Step Out. Esta opção (metaforicamente, **saltar fora!**) apresenta duas “aplicações”.

a) Saltar sobre as máquinas (*stepping over the machines*). Quando a seta de execução está sobre uma submáquina; ao pressionar o botão **step out**, a máquina será executada sem alterar a visualização dessa máquina;

b) o *stepping out* propriamente dito. O outro caso é quando estamos numa máquina e queremos executar o programa até que a máquina termine – é o *stepping out* propriamente dito!

Esta funcionalidade está também disponível tanto em modo de depuração visual como em modo normal.

Opção Step Back. Esta opção permite retroceder passo a passo na execução – é uma verdadeira “retro execução” (*reverse execution*). Está disponível tanto em modo de depuração visual como em modo normal

*Exemplos
de Máquinas*

Problemas de Teoria da Computação

A Teoria da Computação fornece um conjunto de exemplos de Problemas que podemos chamar de “clássicos” e que variam segundo o grau de complexidade teórica. Referimo-nos a “problemas computáveis”, ou seja, resolúveis segundo os princípios de funcionamento de uma Máquina de Turing.

As computações que aqui deixamos simplesmente invocadas e que são, naturalmente, conceptualizáveis e implementáveis através de um Programa-Gráfico em VT são, por exemplo:

Máquinas de fazer operações aritméticas simples:

- Máquina de Somar
- Máquina de (operação inversa:) Subtraír
- Máquina de Multiplicar
- Máquina de (operação inversa:) Dividir

Máquina de copiar *strings*

Máquina de empurrar blocos

Problemas de grande nível teórico:

- Decidibilidade

Problemas computáveis interessantes:

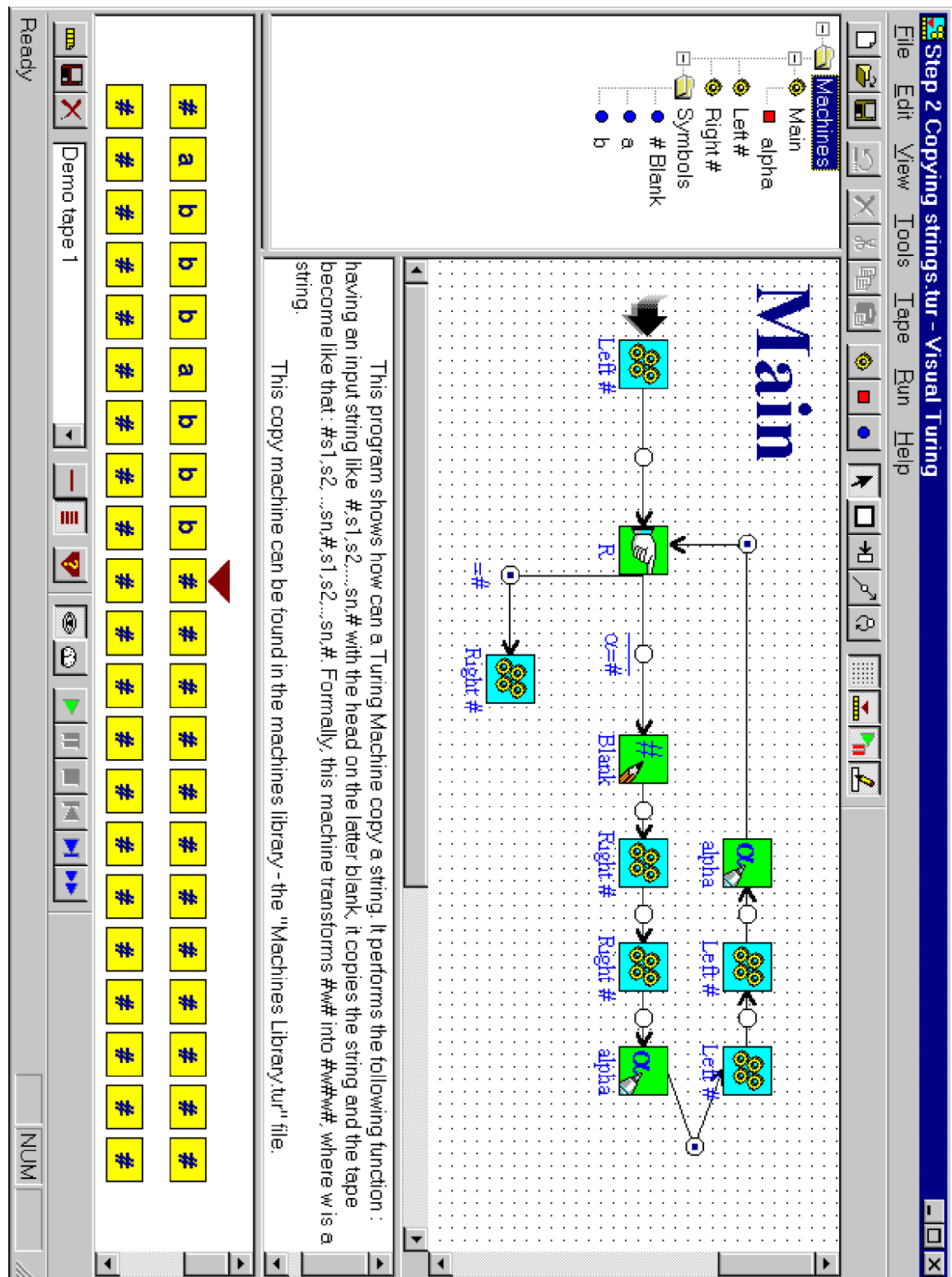
- Semáforo
- Pastor, Lobo, Ovelha, Couve
- Torres de Hanói
- Caixeiro Viajante

UM DESAFIO :

- Euro 2004: como chegar a Campeão

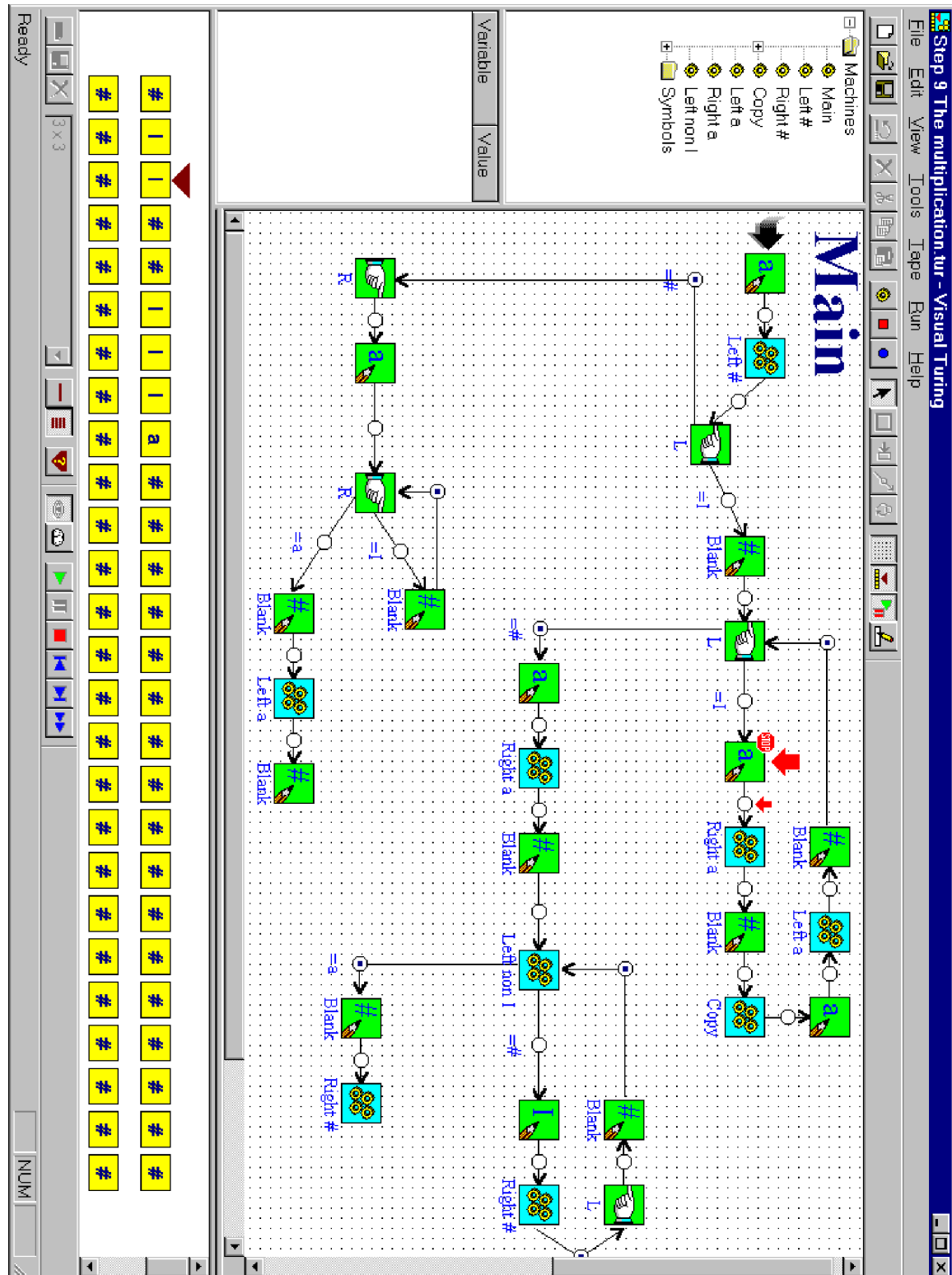
Biblioteca de Exemplos - 1

Uma Máquina que realiza uma tarefa relativamente simples: copiar uma “string”:



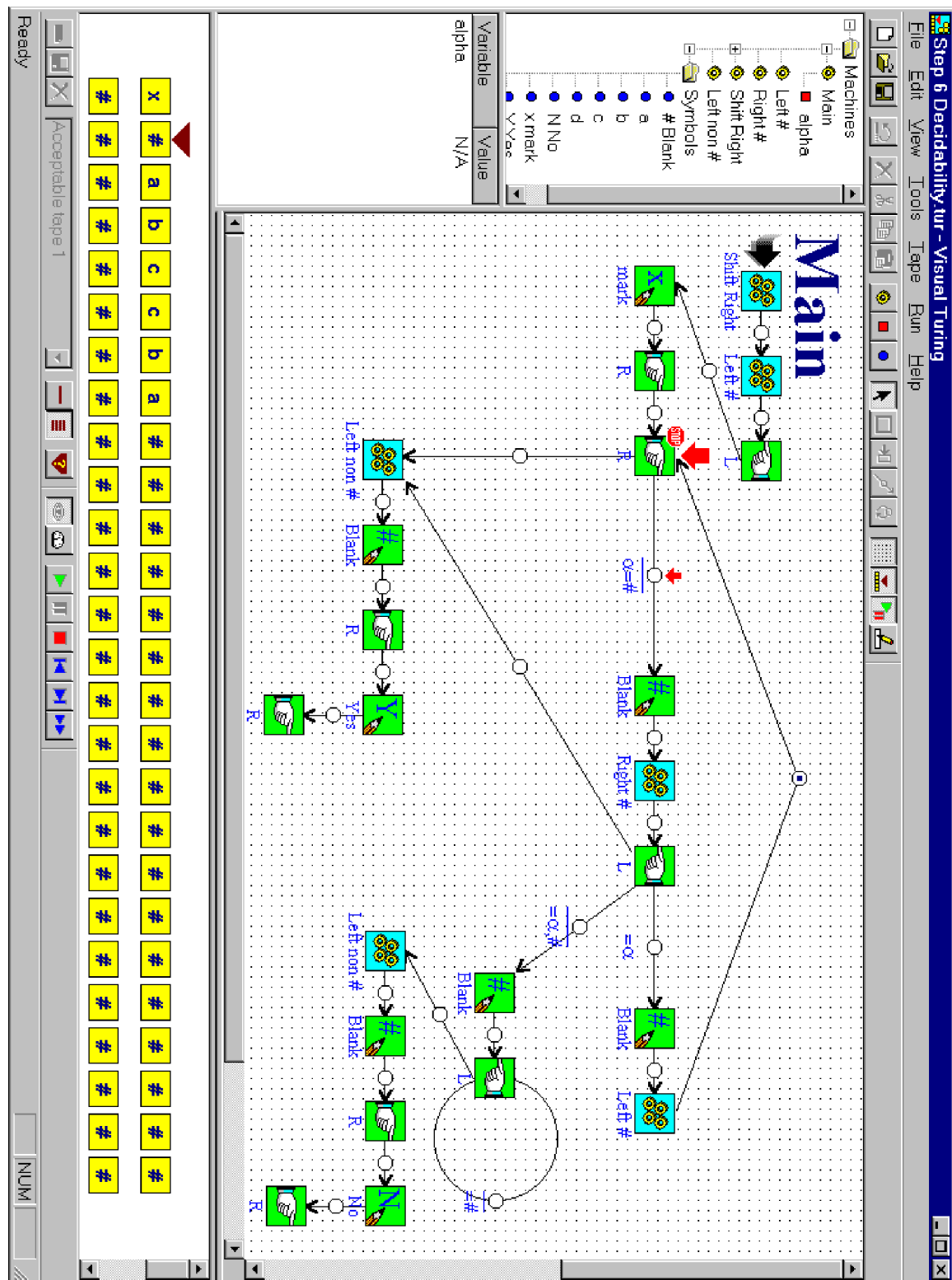
Biblioteca de Exemplos - 2

Uma Máquina de Multiplicar.



Biblioteca de Exemplos - 3

Uma Máquina que verifica a decidibilidade :



Biblioteca de Exemplos - 4

Uma Máquina pensada e implementada no âmbito do Curso de 2004 :

UM DESAFIO: Euro 2004: como chegar a Campeão

0 - INTRODUÇÃO

Foi no âmbito da cadeira de Inteligência Artificial que decidimos realizar este trabalho de aplicação prática dos conhecimentos adquiridos nas aulas.

O EURO 2004 estava no auge e figurou-se como uma válida hipótese de investigação e desenvolvimento de um caso de estudo de IA.

O nosso estudo baseou-se na estrutura de grupos aleatoriamente definida, pelo órgão competente (UEFA), com vista à dissecação de resultados obtidos pelos diversos participantes na fase de grupos, que eram quatro, cada um composto por quatro equipas cada.

Decidimos também, a título introdutório, contextualizar a inteligência artificial, no seguinte ponto.

1 – CONTEXTUALIZAÇÃO DE INTELIGÊNCIA ARTIFICIAL

Apesar de relativamente recente a Inteligência Artificial é a realização de um sonho do homem que remonta à Antiguidade Clássica.

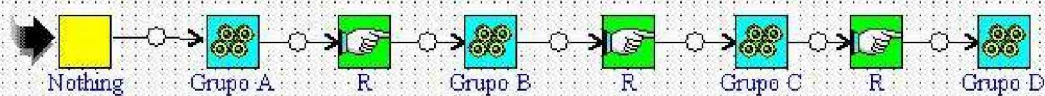
No Renascimento, e com a expansão de um espírito prático e quantitativo, surge a mecânica e, com ela, (e com o aperfeiçoamento do mecanismo do relógio) uma nova concepção do homem.

Os desenvolvimentos em I.A. avançam lado a lado com a evolução dos computadores que, ao longo do tempo foram fazendo com que se comesçassem a encarar essas máquinas como inteligentes alterando mesmo o nosso conceito de inteligência e aproximando os conceitos 'máquina', tradicionalmente não inteligente da 'inteligência', capacidade antes consignada exclusivamente ao homem.

No entanto o seu objecto de estudo continua rodeado de uma certa bruma, no sentido em que o homem ainda não possui uma definição suficientemente satisfatória de inteligência e para se compreenderem os processos da inteligência artificial e da representação do conhecimento terão de se dominar os conceitos de inteligência humana e conhecimento.

Devemos então fornecer à máquina uma avalanche de dados, teorias formais de 'bom senso', de crenças, de um universo simbólico superior ou, pelo contrário, basear o estudo da cognição no nível inferior da percepção e do controlo motor. A tendência geral foi no sentido de conciliar as duas teorias numa terceira teoria híbrida, segundo a qual a máquina seria capaz de raciocinar utilizando conceitos complexos, e de perceber o seu meio envolvente.

Euro 2004



Esta máquina, denominada Euro 2004, é composta por quatro máquinas, Grupo A, Grupo B, Grupo C, Grupo D.

Este programa inicia a sua actividade na máquina Euro 2004, passando depois para o Grupo A, depois de analisados os valores da fita correspondente ao grupo A, vai passar para o Grupo B, Grupo C e finalmente o Grupo D, máquinas estas que são compostas por 4 máquinas respectivamente, que são os participantes.

3 – GRUPO A, GRUPO B, GRUPO C e GRUPO D

Cada uma destas máquinas é composta por quatro máquinas, correspondendo aos participantes de cada Grupo.

Grupo A



Esta máquina do Grupo A tem como participantes Portugal, Grécia, Espanha e Rússia.

Grupo B



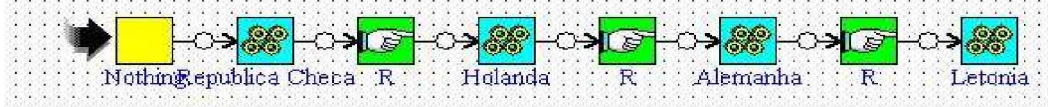
Esta máquina do Grupo B tem como participantes a França, Inglaterra, Croácia e Suíça.

Grupo C



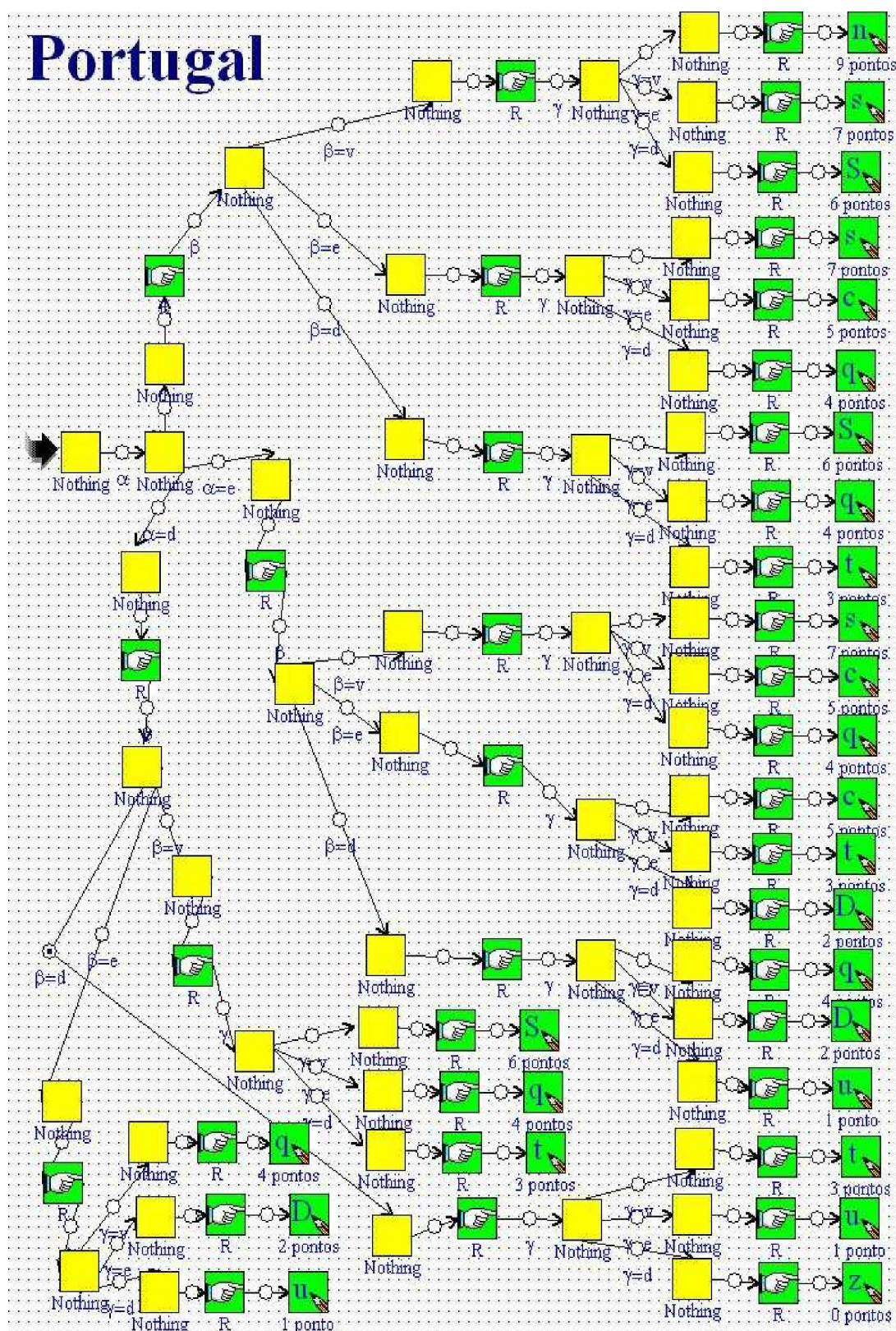
Esta máquina do Grupo C tem como participantes a Suécia, Dinamarca, Itália e Bulgária.

Grupo D



Esta máquina do Grupo D tem como participantes a Republica Checa, Holanda, Alemanha e a Letônia.

A máquina do Grupo A inicia a sua actividade na máquina de Portugal, a qual tem a seguinte função:



3.1 - MÁQUINA PORTUGAL

Nesta máquina é avaliada as primeiras três variáveis existentes nas primeiras três posições da fita, e de acordo com esta avalia-as escrevendo seguidamente na quarta posição da fita o número de pontos obtidos.

Acabando de fazer este processo volta à máquina do Grupo A, vai uma posição para a direita e vai para a máquina seguinte que é a máquina da Grécia, e assim sucessivamente.

Acabando todo o processo inerente ao Grupo A, volta novamente à máquina do euro 2004 vai uma posição para a direita e vai para o Grupo B, e assim sucessivamente até ao Grupo D.

4 - PARTICIPANTES

O início do funcionamento das máquinas (cada participante) começa na instrução **start**, em que depois tem um alfa, o qual assume o valor da primeira posição da fita (V, E ou D, em que V – é vitória, E – Empate e D - Derrota), depois analisa alfa em função de V, E ou D.

No próximo passo vai uma posição para a direita, e logo temos um beta que assume o valor da segunda posição da fita (V, E ou D). Seguidamente, analisa isto outra vez e vai para outra posição, à direita.

Por fim, temos um Y que vai assumir o valor da terceira posição da fita, valor este que depois de analisado vai para a direita e substitui a quarta posição da fita, que tem cardinal, pelo valor de pontos correspondente.

Nota importante:

A fita tem de ser obrigatoriamente preenchida da seguinte forma:

As primeiras três posições são para os valores (V, E ou D), e a quarta posição tem de ficar com cardinal, depois a quinta, sexta e sétima posição têm de ter novamente valores (V, E ou D), e a 8ª com cardinal e assim sucessivamente. Se a fita não obedecer a estes critérios a máquina não funcionará.

Este processo lógico de verificação de todas as possibilidades de resultados, repete-se para todos os participantes de cada grupo (máquinas), como tal, para não estarmos a repetir informação, decidimos não descrever os outros participantes.

5 - VARIÁVEIS

No quadro abaixo podemos ver as variáveis que foram utilizadas para quantificar e qualificar os resultados dos jogos possíveis na fase de grupos para o apuramento dos “quartos de final”.

n	9 pontos
s	7 pontos
S	6 pontos
c	5 pontos
q	4 Pontos
t	3 pontos
D	2 pontos
u	1 ponto
z	0 pontos
v	Vitória
e	Empate

6 – CONCLUSÃO

Foi com imenso agrado que abraçamos o estudo da Inteligência Artificial, pois este revelou-se de suma importância para a nossa maturação académica.

É claro que inicialmente, sentimos algumas dificuldades em estruturar o raciocínio lógico necessário para abordarmos o tema, mas depressa se diluíram.

Foi com imensa pena que decidimos não desenvolver o resto do estudo dos jogos do campeonato, quartos de final, meias finais e final por falta de tempo e alguma complexidade dos mesmos, no entanto estamos convictos dos muitos conhecimentos adquiridos nas aulas e no desenvolvimento e pesquisa para a realização deste trabalho.

Fontes de Informação

Fontes Electrónicas e URLs

Alguns sítios principais sobre Máquinas de Turing e Visual Turing:

Sobre **Turing e Máquinas de Turing**:

<http://www.turing.org.uk/turing/> The Alan Turing Homepage; criada e mantida por um ilustre professor de lógica matemática, Andrew Hodges;

<http://www-csli.stanford.edu/hp/Turing1.html> Uma excelente página sediada na Universidade de Stanford sobre Máquinas de Turing;

<http://www.ams.org/new-in-math/cover/turing.html> Uma excelente página da American Mathematical Society, sobre Máquinas de Turing.

Sobre **Visual Turing**:

<http://www.cheransoft.com/vturing/> Página do autor do Visual Turing evocado neste Manual, Cristian Cheran.

<http://www.cheransoft.com/vturing/download.html> ...para o download de VT

Fontes Bibliográficas

Há uma imensidão de fontes bibliográficas. Recomenda-se que sigam as sugestões bibliográficas das páginas electrónicas anteriores, mais as listas simples e cruzadas de obras encontráveis em www.amazon.com sob as palavras chave

- (1) Turing,
- (2) Máquinas de Turing e
- (3) Visual Turing.