Default

COLLABORATORS			
	TITLE : Default		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 9, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Default **1**

1.1 Deduce 1.22 . . . 1
1.2 Disclaimer . . . 1
1.3 What is Deduce? . . . 1
1.4 Installation . . . 2
1.5 Getting Started . . . 2
1.6 noreqtools . . . 2
1.7 Using Deduce . . . 2
1.8 Commands . . . 3
1.9 Load . . . 3
1.10 Save . . . 4
1.11 Why . . . 4
1.12 Help . . . 4
1.13 Quit . . . 4
1.14 Divulge . . . 5
1.15 Forget . . . 5
1.16 Some notes about the program . . . 5
1.17 Examples to get you started . . . 5
1.18 To Do . . . 6
1.19 Known Problems . . . 6
1.20 History . . . 6
1.21 Contacting the Author . . . 8
1.22 Acknowledgements . . . 8

Chapter 1

Default

1.1 Deduce 1.22

```
  Deduce! version 1.22
by James Williams and Eric Augustine

      Disclaimer
    What is Deduce?
     Installation
    Getting Started
     Using Deduce
       Commands
Some notes about Deduce
       Examples
        To Do
    Known problems
       History
 Contacting the author
   Acknowledgements
```

1.2 Disclaimer

This program is freeware. It is being distributed with no warranty.
I can't be held responsible for any damages caused either directly or
indirectly by the use of this program. Use it at your own risk.

This program cannot be sold for profit. If you wish to use this program
in a commercial distribution, you must first obtain written permission from
me.

1.3 What is Deduce?

Deduce is an experiment in artificial intelligence. It attempts to simulate
deductive reasoning. Consider the following example:

Spot is a dog.

A dog is an animal.
Therefore, Spot is an animal.

Deduce will accept input in the form of English sentences and upon request will answer yes/no questions about what it has been taught.

1.4 Installation

As of right now, Deduce doesn't have an install script, so you will have to do the installation manually. Just copy the ReqTools library to your libs: directory if you don't already have one there, and then drag Deduce's drawer icon to where ever you want to put it. That's it! Hopefully I will have an install script of some kind by the next release.

1.5 Getting Started

Deduce must be run from the CLI. Open a CLI or Shell window, cd to the directory with Deduce, and type Deduce. At this point, you should see the program name, my name, and a prompt waiting for input.

The syntax for Deduce is:

Deduce [options]

The options are:

noreqtools

For those of you who prefer to use the Workbench, I've included a script so you can run Deduce (with no options) just by double clicking on the icon.

1.6 noreqtools

The noreqtools option will disable the use of the ReqTools interface in favor of the standard CLI interface.

1.7 Using Deduce

Most of the time, you will probably be entering statements and asking questions. I line of input has the form:

subject [helping verb] [verb] object

You can also make a statement negative by including the word 'no'. For example, you could say "A dog is not a cat." In fact, the only way you can ever get a no response to a question is by entering negative

statements.

Deduce interprets any statement beginning with a "question" word to be a question. If Deduce is unable to find a definite answer to your question, it will respond with "I don't know". Sometimes it may seem like Deduce should know the answer to a question, but it says it doesn't know. Consider the following example:

```
A liquid will evaporate.
A solid is not a liquid.
Will a solid evaporate?
```

It looks like Deduce should be able to figure out that since a solid is not a liquid, it won't evaporate. However, logically, there isn't enough information given to make that conclusion. Deduce doesn't know anything about a solid, and so it doesn't know for sure that it can't evaporate. Perhaps another example will make this clearer.

```
Water will evaporate.
Alcohol is not water.
Will Alcohol evaporate?
```

This time it should be clear that simply because alcohol is not water doesn't mean it won't evaporate.

When you enter something, Deduce knows the opposite isn't necessarily true. So if you enter "A cat is not a dog", Deduce won't automatically know that a dog is not a cat.

1.8 Commands

Aside from entering statements and questions, you can also tell Deduce to do something for you. These are the commands it understands.

```
   Load
   Save
   Why
   Help
   Quit
  Divulge
  Forget
```

1.9 Load

Load:

Load will load in a previously saved session and add it to what's currently in memory. Load is entered on a line by itself, afterwhich you will be presented with a ReqTools interface to select a file. If there isn't enough memory, you will have to enter the filename at a standard CLI prompt.

Note : This doesn't clear the current memory. If you need to erase what's
already in memory, use the forget command.

1.10 Save

Save:

To save your work, type save, then select your save-filename using the ReqTools interface. If you have insufficient memory, you will have to enter the filename at a CLI prompt.

Save will create a text file containing all the statements you've entered. If you want to change something you've entered, you can save your work and edit the save file with any text editor or word processor than can load and save plain ascii files. An ampersand (&) marks the end of input, so you can put comments after it.

1.11 Why

Why:

The why command is usually used to let you see the logic behind one of Deduce's responses to a question. You must first ask the question, then enter "why" on a line by itself to see Deduce's reasoning.

You may also use the why command if you enter a statement and Deduce responds with "Yeah, I know" or "That can't be right...". These messages mean you tried to enter something that Deduce already knows. Typing "why" at the next prompt will show you how the information is known.

1.12 Help

Help:

The help command will present you with a list of the commands available and give you a help prompt. Entering the name of one of the commands will give you more detailed help on that command, then exit you out of the help mode. Thanks to Eric for writing this command.

1.13 Quit

Quit:

When you get tired of playing with Deduce, type "quit" to leave the program. It is synonymous with exit, bye, stop, and end.

1.14 Divulge

```
Divulge:

Divulge will print the entire contents of Deduce's database.  This was
originally put in for debugging purposes, but I decided to leave it in.
```

1.15 Forget

```
Forget:

Forget makes Deduce empty its database.  This can be useful in conjunction
with the load command if you want to completely replace what's in Deduce's
memory.
```

1.16 Some notes about the program

```
Deduce tries to interpret your sentence by looking at the position of the
words.  Therefore, it is important to follow a few rules when teaching
Deduce new information.  Your subject and object nouns need to be
one word.  If you have to use more that one word (for an adjective or
whatever) put an underscore character between the words.  This doesn't apply
to articles (a, an, the) and other known words (all, every, ever, of).
Deduce understands negative contractions (don't, doesn't, isn't, aren't,
won't, can't, wasn't, weren't, ain't) as well as no and not.  Deduce isn't
case sensitive, so don't worry about matching case on everything.  Deduce
automatically strips out all punctuation from a sentence.  Words are limited
to 20 characters, and sentences are limited to 20 words and 100 characters.
Anything more will be stripped off.
```

1.17 Examples to get you started

```
Here are some examples you can try to get you started.

Spot is a dog.
A dog is an animal.
A dog will eat dog_food.
Chunkies is a dog_food.
A cat is an animal.
A dog is not a cat.
Will Spot eat Chunkies?
why
A horse is an animal.
A horse will eat hay.
A horse will drink water.
A horse will not eat meat.
A square is a rectangle.
A rectangle is a polygon.
A liquid will evaporate.
```

```
Water is a liquid.
Alcohol is a liquid.
```

1.18 To Do

```
- The parser should be able to understand more complex sentences, and be
  able to answer more complex questions (besides the yes/no type).  I
  have some ideas in mind, but I don't know how many of them I will be
  able to implement, so I don't want to say too much just yet.

- Sooner or later I will probably have to build a GUI for this program.
  I've never actually written a GUI before, so give me some time.

- Build a configuration file.

- Buy an 030 + MMU + memory expansion for my 1200 so, among other things,
  I can run enforcer to check my program.  (If anyone finds that this
  program causes enforcer hits, be sure to let me know)
```

1.19 Known Problems

```
- The parse relies on the position of words to determine their part of
  speech, and can therefore easily misinterpret your input.
  e.g. "Green is a good color" <- good will be interpreted as a verb
  (good color should have been written as good_color)

- Memory is allocated in small chunks.  This can lead to memory fragmentation
  (fragmentation should be minimal unless you build a huge database)

- When asking a question, negative words are ignored, so if you entered
  "Is a dog not a cat", it would be interpreted as "Is a dog a cat".

- Sometimes when you ask "why" to a question, the statements will appear in
  a somewhat scrambled order.  I know what's causing this, but fixing it
  would be a bit messy.  And besides, it's not THAT bad, is it?

- Since my last upload of Deduce to Aminet, I've had problems with
  the ReqTools interface.  Typing load returns "load cancelled" and typing
  save returns "save cancelled".  If anyone can figure out what's causing
  this (perhaps a memory leak? Enforcer users?), let me know.  If you

  have this problem, you can get around it with the noreqtools option: ie

  Deduce noreqtools
```

1.20 History

```
History:

Jan 31, 1995 : version 1.00
```

```
    First release

Feb 3, 1995 :  version 1.10
  * Added load and save commands
  * Made an AmigaGuide version of the documentation

Mar 14, 1995 : version 1.20
  * I accidently limited filenames to 20 characters on the last version.
    Sorry 'bout that.
  * Fixed a bug that would sometimes cause Deduce to respond "I don't know"
    to a question it should know.  For example:
      A horse will not eat meat.
      Pork is a meat.
      Will a horse eat pork?
  * Added some simple online help (actually, Eric added it)
  * Added a forget command to empty Deduce's database
  * Fixed a bug that caused segmentation faults on a Sun Sparc Station
    (This shouldn't affect the Amiga version)
  * Finally broke down and bought SAS C/C++ 6.51. (That's right - I should
    be able to compile that GUI code now ;) )  Since then, I've patched it
    to 6.55.
  * Added code to convert between first and second person.  For example:
      >You are a computer
      Ok.
      >Are you a computer?
      Yes
      >why
      Because:
       I am a computer
    (thanks to Hendrik Fuss for the suggestion)
  * Fixed a bug that caused Deduce problems if you typed 'why' twice in a
    row
  * Added a ReqTools interface for loading and saving
  * Deduce now uses variable length records for storage, resulting in better
    memory conservation

March 17, 1995 : version 1.21 (bugfix)
  * When converting between 1st and 2nd person, Deduce would extract words
    from inside other words: e.g. it would see "you" in "youth" and replace
    it with "Ith" (assuming the sentence was told in second person)
    (thanks to Arthur Hagen for pointing this out)
  * I overlooked a special case with negative logic.  Take for example:
      >I hate spam
      >Vegetables are not spam
      >Do I hate vegetables?
      No
    This has been fixed
    (again, thanks to Arthur Hagen for pointing this out)
  * Added a noreqtools option for the command line.

April 14, 1995 : version 1.22 (bugfix)
  * Fixed that enforcer hit.  See up on changes for version 1.20 where
    it says "Fixed a bug that caused segmentation faults on a Sun
    Sparc Station" ?  Well, that code somehow got lost and wasn't
    included in the Amiga version.  And when I upgraded to SAS/C, it
    DID start causing problems. (Thanks Stefan Dube and Simon Stelling)
  * Fixed another problem with negative logic.  Consider the following:
```

```
      a dog will eat dog_food
      a cat is not a dog
      will a cat eat dog_food?
    Deduce would answer yes.  This has been fixed. (Thanks Stefan Dube)
  * Deduce no longer prints the blank lines in its output. (suggested by
    Simon Stelling)
```

1.21 Contacting the Author

If you need to contact me (for comments, questions, suggestions, bug reports, etc), my email address is :

    williamj@griffon.mwsc.edu

If you don't have email, my address is:

    James Williams
    209 Thomas
    Weston, MO 64098 USA

And, if you're lucky, you may even catch me on the #amiga channel on IRC with the name JamesW.

1.22 Acknowledgements

I would like to thank the following people who helped make Deduce a reality:

The authors of Mind.rexx (util/rexx/Mindrx.lha on Aminet). It was what inspired me to write Deduce.

Ronald Carnell, author of "Smart Alec" (Compute! Sept. 1987). He supplied the basic algorithm used in "Smart Alec", which I took and expanded upon to create Deduce.

Eric Augustine for helping me to write Deduce. He wrote the code for the help command and has offered to help with the GUI.

Nico Francois for writing the fantastic ReqTools requestor library.

Everyone who has uploaded programs to the Aminet, as well as Urban Mueller for creating the Aminet. Without such a large source for Amiga software, the Amiga would most likely have faded away a long time ago.

And everyone who owns and supports our beloved Amiga computer. I hope to see Amiga business pick up after this #$%?! bidding is over. Keep up the good work, everyone!