

A Machine Translation Breadboard

Martin Kay

Stanford University
and the University of the Saarland

Ruby

Documentation for version 1.9

<http://www.ruby-doc.org/core-1.9/index.html>

Another Reference

<http://www.rubyist.net/%7eslagell/ruby/>

The Standard Text by Dave Thomas

<http://www.ruby-doc.org/docs/ProgrammingRuby/>

Changes in Version 1.9

<http://eigenclass.org/hiki.rb?Changes+in+Ruby+1.9>

Classes

```
class Foo
  def initialize(a, b, c=nil)
    @a = a
    @@b = b
    @c = c
  end
  def get_a
    @a
  end
  if $debugging
    def get_a
      puts "a = #{@a}"
      @a
    end
  end
end
```

end
Martin Kay

Must be a constant

alpha	#	local
@beta	#	object
@@gamma	#	class
\$delta	#	global
Epsilon	#	constant

Classes

```
class Foo < Fie
  ...
end
```

```
class String
  def to_utf8
    ...
  end
end
```

```
class Integer
  def double
    self + self
  end
end
```

6.double
--> 12

Iterators

```
def scan_tree
  yield self
  left.scan_tree{|x| yield x} if left
  right.scan_tree{|x| yield x} if right
end

tree.scan_
tree.scan_
  foo(st)
end
```

```
def scan_tree(&p)
  p.call(self)
  left.scan_tree(p) if left
  right.scan_tree(p) if right
end
```

```
def scan_tree(&p)
  ...
  right.scan_tree(&p) if right
  ...
end
```

Examples

```
class TaskBase

  attr_reader :link, :number

  include Monitorable

  @@task_number = 0

  def initialize(monitor)

    @link = nil
    @monitor = monitor
    @number = @@task_number += 1

  end
```

Examples

```
def scan_tasks

  lnk = get_link
  while lnk && lnk.deleted?
    puts_if 'task', "          Deleted #{lnk}"
    put_link(lnk = lnk.get_link)
  end
  lnk.scan_tasks{|t| yield t} if lnk
  yield self
end
```

Examples

```
class Limit < TaskBase

  include Comparable

  ...

  def <=>(other)

    @opt <=> other.opt

  end
```



```
class GramTask < ActiveSupportLR
  ...
  def compatible(other)
    @dest == other.org && \
      (other.kind_of?(MorphemeTask) \
       || other.kind_of?(PhraseTask))
  end
  def exec(task, chart)
    return unless compatible(task)
    ...
  end
end
```

Examples

```
class OptimalityAgenda < Agenda
  def put(task)
    if reject(task)
      return nil
    end
    j = 0
    @members.length.times do |i|
      if task.opt <= @members[i].opt
        task, @members[i] = @members[i], task
      else
        j+=1
      end
    end
    @members << task
    return j
  end
end
```

chart

```
schedule(task)
work_agenda(block=nil)
apply(active, inactive)
get_vertex(name)
get_state(v, s=nil)
each_vertex
sorted_vertices
show(detail=false)
```

German Morphgraphemics

```
define coda      [ [ 0 <- space || .#. _ ] .o.  
                  [ space <- 0 || _ .#. ] ];  
define consonant [ b | c | d | f | g | h | j | k | l | m | n | p |  
                  q | r | s | t | v | w | x | y | z ] ;  
define vowel     [ a | e | i | o | u ] ;  
define nonuml    alpha - [ a | o | u ] ;  
define boundary  [ .#. | % + ] ;  
define optional  [ %? (->) 0 ] ;  
  
define umlautA1  [ a -> ä || _ (u) nonuml+ EM "^" ] ;  
define umlautA2  [ "^" -> 0 || ä (u) nonuml+ EM _ ] ;  
define umlautOU  [ o -> ö, u -> ü || _ nonuml+ EM "^" ] ;  
define umlaut    [ "^" -> 0 ] ;  
define noumlaut  [ ä (->) a e ] .o. [ ö (->) o e ] .o. [ ü (->) u e ] ;  
define Einsertion1 [ [..] -> e || [d | t] EM _ (s) t EM ] ;  
define Einsertion2 [ [..] -> e || [consonant - [l | r]] m EM _ (s) t EM ] ;  
define Edeletion  [ e -> 0 || [e ([l | r])] EM _ (e) n EM ] ;  
  
Define DiacriticDeletion [ diacritic -> 0 ] ;  
define BoundaryDeletion [ [BM | EM] -> 0 ] ;
```

```
Define Word [ [ preamble .o.  
                optional .o.  
                umlautA1 .o.  
                umlautA2 .o.  
                umlautOU .o.  
                umlaut .o.  
                noumlaut .o.  
                Einsertion1 .o.  
                Einsertion2 .o.  
                Edeletion .o.  
                DiacriticDeletion .o.  
                BoundaryDeletion] | 0 ];
```

```
read regex [Word [space Word]* ] ;
```

```
name net de ;
```

```
write prolog >mg_de.pr ;
```

```

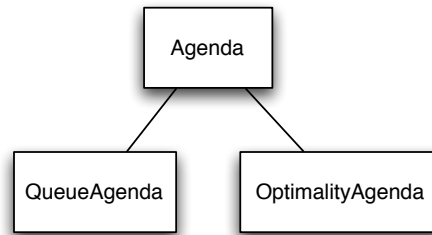
# -*- coding: iso-8859-1 -*-
network(de).
symbol(de, "BM").
symbol(de, "EM").
symbol(de, "BSLASH").
symbol(de, "BB").
symbol(de, "EB").
arc(de, 0, 1, "?").
arc(de, 0, 2, "a").
arc(de, 0, 3, "b").
arc(de, 0, 3, "c").
arc(de, 0, 4, "d").
...
arc(de, 27, 61, "ö").
arc(de, 27, 61, "ü").
arc(de, 27, 67, "u":"ü").
arc(de, 27, 68, "\\":"{").
arc(de, 27, 69, "\\":"}").
arc(de, 27, 70, "ä":"a").
arc(de, 27, 70, "ö":"o").
arc(de, 27, 70, "ü":"u").
arc(de, 28, 1, "?").
arc(de, 28, 2, "a").
...
arc(de, 69, 61, "}":"0").
arc(de, 70, 115, "0":"e").
arc(de, 70, 70, "%?":"0").
arc(de, 70, 70, "^":"0").
arc(de, 71, 71, "?").
...

```

translator

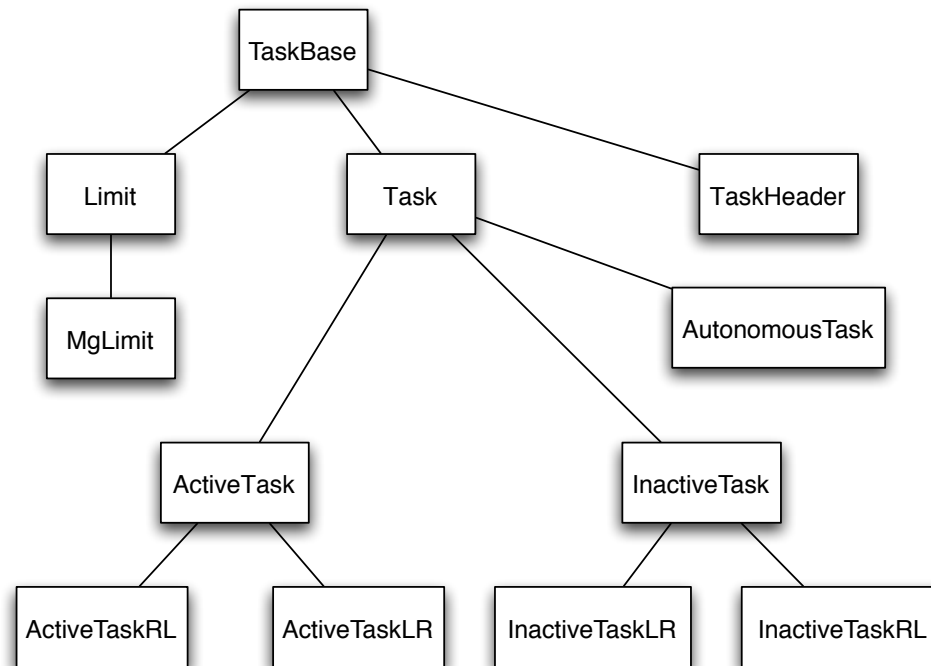
chart

agenda



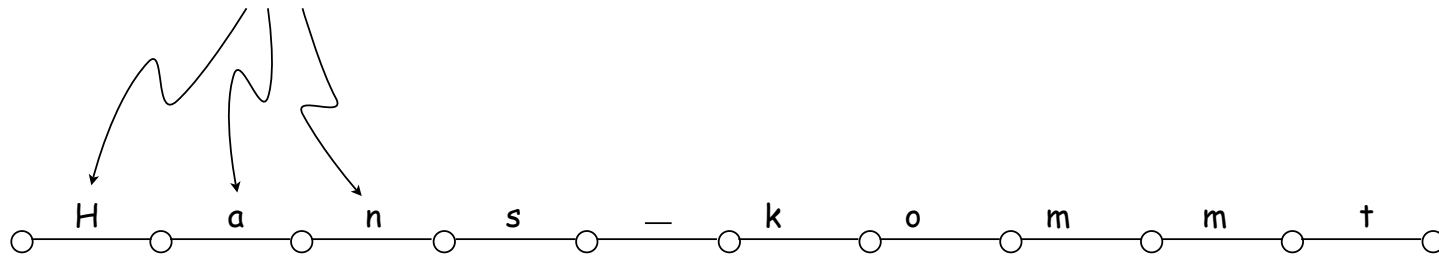
Tasks

Task = Edge

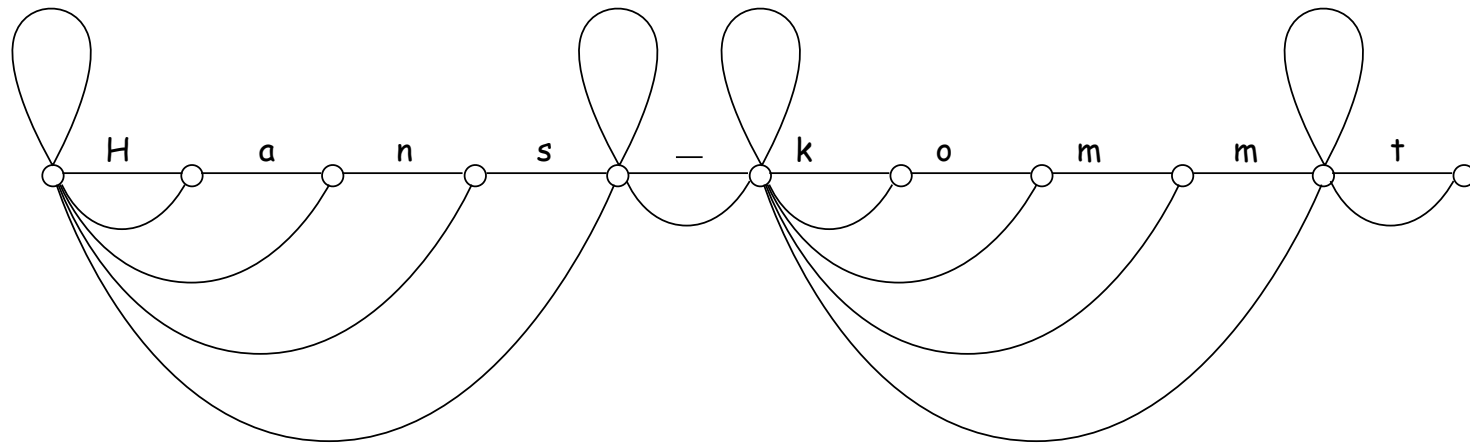


Morpheme Recognition

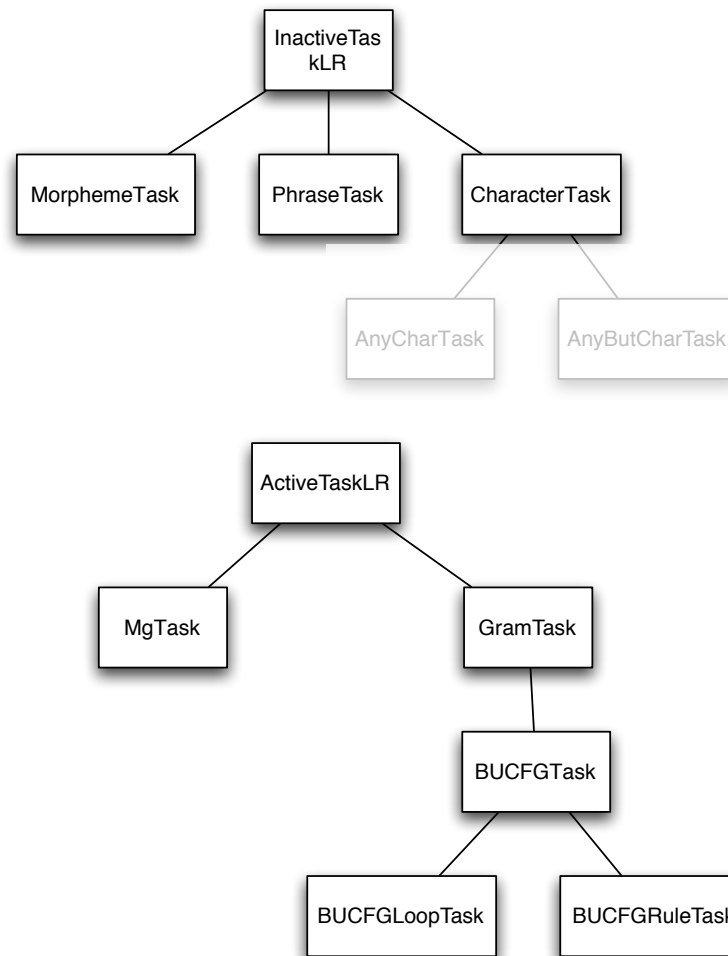
Inactive
Character Tasks



Morpheme Recognition



Grammar Tasks



Dictionary



Files

AttVal.rb~
CFGGrammar.rb
GermanAVGrammar.rb
GermanCFGGrammar.rb
GermanTranslator.rb
Grammar.rb
Grammar.rb~
OptimalityMarks.rb
OptimalityStatus.rb
OptimalitySystem.rb
OptimalityVector.rb
Parser.rb
Project.tproj
TestAVGTasks.rb
Utilities

agenda.rb
character_task.rb
chart.rb
config
config_test.rb
de_lex.pr
dictionary.rb
dictionary_entry.rb
en_lex.pr
globals.rb
mg_de.fsm
mg_de.pr
mg_en.fsm
mg_en.pr
mg_task.rb
morpheme_task.rb

prolog_fsa.rb
status.rb
tasks.rb
test_att_val.rb
test_chars.rb
test_de_mg
test_de_mg~
test_fsm~
utf8.rb
vertex.rb

Utilities

cmd_line.rb
extensional_fsa.rb
fs_root.rb
mon.rb
monitorable.rb
op_prec.rb
printer.rb

Unit testing

/Users/kay/rubyunit-0.5.6/runit/cui/testrunner
/Users/kay/rubyunit-0.5.6/runit/testcase
/Users/kay/rubyunit-0.5.6/runit/testsuite