

RA: a new engine for NooJ

max.silberztein@univ-fcomte.fr

The engine, v1.0

- Open source
- Developed with Swift, compatible with MacOS, Linux, Unix, Windows
- 19 modules (dictionary, orthographic and syntactic grammars...)
- Test-Driven Development (9,000 lines of test code)
- Managed by a version control system (git)
- Toolbox and examples downloadable at www.nooj4nlp.org
- Collaborative, source available at:
<https://gitlab.com/Silberz/ra-linguistic-engine>

The source

The screenshot shows the GitLab web interface for the repository 'ra-linguistic-engine' by Max Silberstein. The browser address bar shows the URL 'https://gitlab.com/Silberz/ra-linguistic-engine'. The left sidebar contains a navigation menu with options like Project information, Repository, Issues, Merge requests, CI/CD, Security & Compliance, Deployments, Packages & Registries, Infrastructure, Monitor, Analytics, Wiki, Snippets, and Settings. The main content area displays the repository details, including the name 'ra-linguistic-engine', Project ID '31651161', and statistics: 48 Commits, 1 Branch, 0 Tags, and 26 MB Project Storage. Below this, there's a section for the latest commit 'added concordance' by Max Silberstein, authored 21 hours ago, with a commit hash '7ab1b40d'. A list of files and their last commit details is shown in a table.

Name	Last commit	Last update
LinguisticResources	added en-PrepNg.nog	2 days ago
RA.xcodeproj	added concordance	21 hours ago
RA	added concordance	21 hours ago
RATests	ALU: added various checking and parsing	2 days ago
concordance	added concordance	21 hours ago
dic2lst	Added lst2ra nog2ra nom2ra printra prints...	2 days ago
lexicalanalysis	Added lst2ra nog2ra nom2ra printra prints...	2 days ago
lst2ra	Added lst2ra nog2ra nom2ra printra prints...	2 days ago
nog2ra	Added lst2ra nog2ra nom2ra printra prints...	2 days ago
nom2ra	Added lst2ra nog2ra nom2ra printra prints...	2 days ago
printra	Added lst2ra nog2ra nom2ra printra prints...	2 days ago

Linguistic resources

.dic : dictionary, i.e. list of Atomic Linguistic Units (ALUs)

.idx : an index of matches with corresponding outputs

.lst : a list of forms linked to their ALUs

.nof : inflectional / derivational grammar

.nog : syntactic grammar

.nom : morphological / orthographic grammar

.not : a text + a Text Annotation Structure (TAS)

.ra : a Multiple Generalized Finite-State Automaton (MGFSA)

RA is 95% compatible with NooJ but...

- All automata are Multiple-Generalized-Finite-State-Automata (MGFSA)
- Dictionaries, orthographic, morphologic and syntactic grammars are compiled into MGFSA
- All MGFSA are stored in the same file format : **.ra**
- Only one format for all types of dictionary (DELAS, DELAF and DELAV):
 - NooJ: **eating,eat,V+G**
 - RA: **eating,V+G+_LEX=""**
- Text can be segmented into text units (e.g. sentences) with orthographic grammars
- A new collocation operator for grammars, e.g. “pizzeria” terms in the context of “Clinton”:
Pizzagate = <pizzeria> & Clinton ;
- All languages’ morphological operators available: Arabic **<M>** operator in NooJ → **<ARM>** in RA
- Special character “**#**” replaced with **<_>**; “**=**” replaced with **<_>**; “**_**” replaced with **<_ ->**
- ...

RA linguistic resources are easier to understand

RA's parsers are more efficient than NooJ's...

- Analysis of agglutination, e.g. a wordform contains 45 letters:

pneumonoultramicroscopicsilicovolcanoconiosis

needs to be analyzed as a sequence of 6 ALUs: <pneumono,PREFIX> <ultra,PREFIX>

<microscopic,PREFIX> <silico,PREFIX> <volcano,PREFIX> <coniosis,N>

- NooJ grammar:

(\$(P <L> <L>* \$)/<\$P=:PREFIX>)* \$(N <L> <L>* \$)/<\$N=:N>

Complex to understand, and not efficient: $O(2^n)$

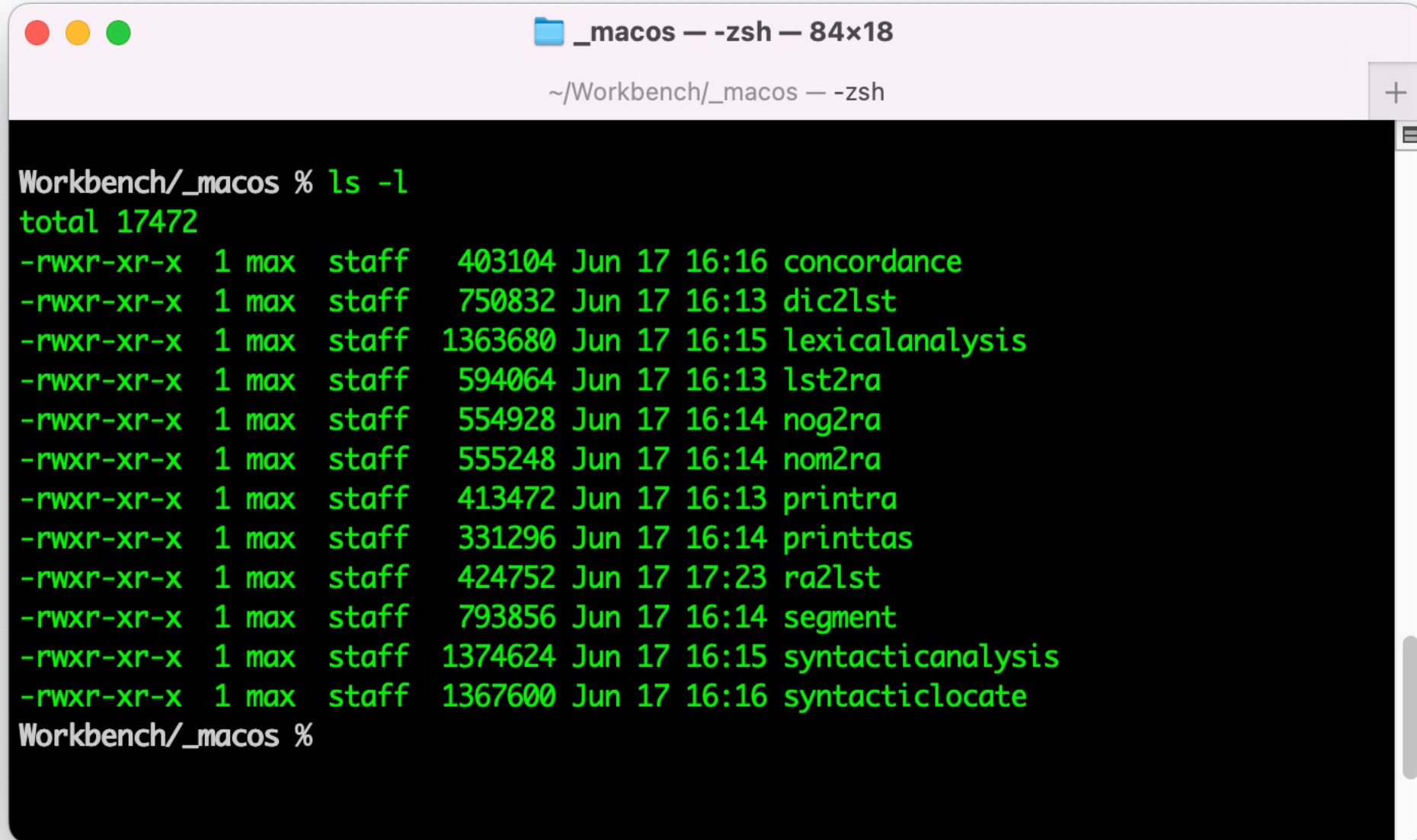
=> NooJ checks $2^{45} = 35,184,372,088,832$ constraints <\$P=:PREFIX>

- RA grammar: **<PREFIX>* <N>**

Easier to understand, and very efficient: $O(n)$

=> RA performs 45 dictionary lookups in worst case

RA toolbox v0.9 contains 12 line commands



```
Workbench/_macos % ls -l
total 17472
-rwxr-xr-x  1 max  staff   403104 Jun 17 16:16 concordance
-rwxr-xr-x  1 max  staff   750832 Jun 17 16:13 dic2lst
-rwxr-xr-x  1 max  staff  1363680 Jun 17 16:15 lexicalanalysis
-rwxr-xr-x  1 max  staff   594064 Jun 17 16:13 lst2ra
-rwxr-xr-x  1 max  staff   554928 Jun 17 16:14 nog2ra
-rwxr-xr-x  1 max  staff   555248 Jun 17 16:14 nom2ra
-rwxr-xr-x  1 max  staff   413472 Jun 17 16:13 printr
-rwxr-xr-x  1 max  staff   331296 Jun 17 16:14 printtas
-rwxr-xr-x  1 max  staff   424752 Jun 17 17:23 ra2lst
-rwxr-xr-x  1 max  staff   793856 Jun 17 16:14 segment
-rwxr-xr-x  1 max  staff  1374624 Jun 17 16:15 syntacticanalysis
-rwxr-xr-x  1 max  staff  1367600 Jun 17 16:16 syntacticlocate
Workbench/_macos %
```

Parameters to set before using RA tools

- Terminal running **zsh**: if you have unzipped and installed the workbench directory as: `/Users/Joe/Workbench`, then add the following two lines inside the `.zshrc` file in your home directory:

```
PATH="/Users/Joe/Workbench/_macos:${PATH}"  
export PATH
```

- Terminal running **bash**: if you have unzipped and installed the workbench in: `/Users/Joe/RA-Workbench`, then add the following line inside the `.bashrc` file in your home directory:

```
export PATH=/Users/Joe/Workbench/_macos:"${PATH}"
```

dic2lst

inputs:

- dictionaries
- inflectional and derivational grammars

output:

- list of all generated inflected and derived forms

Generated forms are linked to the lexeme via the **LEX** special property

```
Workbench — -zsh — 102x40
~/Workbench — -zsh

~/Workbench % dic2lst
Usage: dic2lst dictionary.dic+ morphologicalGrammars.nof* propertiesDefinition.def
Note: some morphologicalGrammars might be referenced from inside dictionaries with the #use command
e.g.: dic2lst en-Nouns.dic en-Verbs.dic properties.def
e.g.: dic2lst en-dictionary.dic Nouns.nof Verbs.nof Derivations.nof properties.def
→ prints list of all generated forms

~/Workbench % dic2lst simplifiedictionary.dic properties.def
aberration constant,N+DOM="Phys"+_LEX="<E>" +FLX=APPLE+Number=s
aberration constants,N+DOM="Phys"+_LEX="<B>" +FLX=APPLE+Number=p
aberration,N+_LEX="<E>" +FLX=APPLE+Number=s
aberrations,N+_LEX="<B>" +FLX=APPLE+Number=p
aberrational,A
Abert's towhee,N+_LEX="<E>" +FLX=APPLE+Number=s
Abert's towhees,N+_LEX="<B>" +FLX=APPLE+Number=p
abesse,N+Distribution=Hum+_LEX="<E>" +FLX=APPLE+Number=s
abesses,N+Distribution=Hum+_LEX="<B>" +FLX=APPLE+Number=p
abets,V+Trans=T+_LEX="<B>" +FLX=BEG+Number=s+Pers=3+Tense=PR
abet,V+Trans=T+_LEX="<E>" +FLX=BEG+Number=p+Pers=123+Tense=PR
abet,V+Trans=T+_LEX="<E>" +FLX=BEG+Tense=INF
abetted,V+Trans=T+_LEX="<B3>" +FLX=BEG+Number=s+Pers=123+Tense=PT
abetted,V+Trans=T+_LEX="<B3>" +FLX=BEG+Tense=PP
abetted,V+Trans=T+_LEX="<B3>" +FLX=BEG+Number=p+Pers=123+Tense=PT
abetting,V+Trans=T+_LEX="<B4>" +FLX=BEG+Tense=G
abet,V+Trans=T+_LEX="<E>" +FLX=BEG+Number=s+Pers=12+Tense=PR
abetment,N+_LEX="<E>" +FLX=APPLE+Number=s
abetments,N+_LEX="<B>" +FLX=APPLE+Number=p
abettal,N+_LEX="<E>" +FLX=APPLE+Number=s
abettals,N+_LEX="<B>" +FLX=APPLE+Number=p
abetter,N+Distribution=Hum+_LEX="<E>" +FLX=APPLE+Number=s
abettors,N+Distribution=Hum+_LEX="<B>" +FLX=APPLE+Number=p
abetting,A
abetting,N+_LEX="<E>" +FLX=APPLE+Number=s
abettings,N+_LEX="<B>" +FLX=APPLE+Number=p
abettor,N+Distribution=Hum+_LEX="<E>" +FLX=APPLE+Number=s
abettors,N+Distribution=Hum+_LEX="<B>" +FLX=APPLE+Number=p
abeyance,N+_LEX="<E>" +FLX=APPLE+Number=s
abeyances,N+_LEX="<B>" +FLX=APPLE+Number=p
# Dictionary successfully generated 28 forms.

~/Workbench %
```


lst2ra

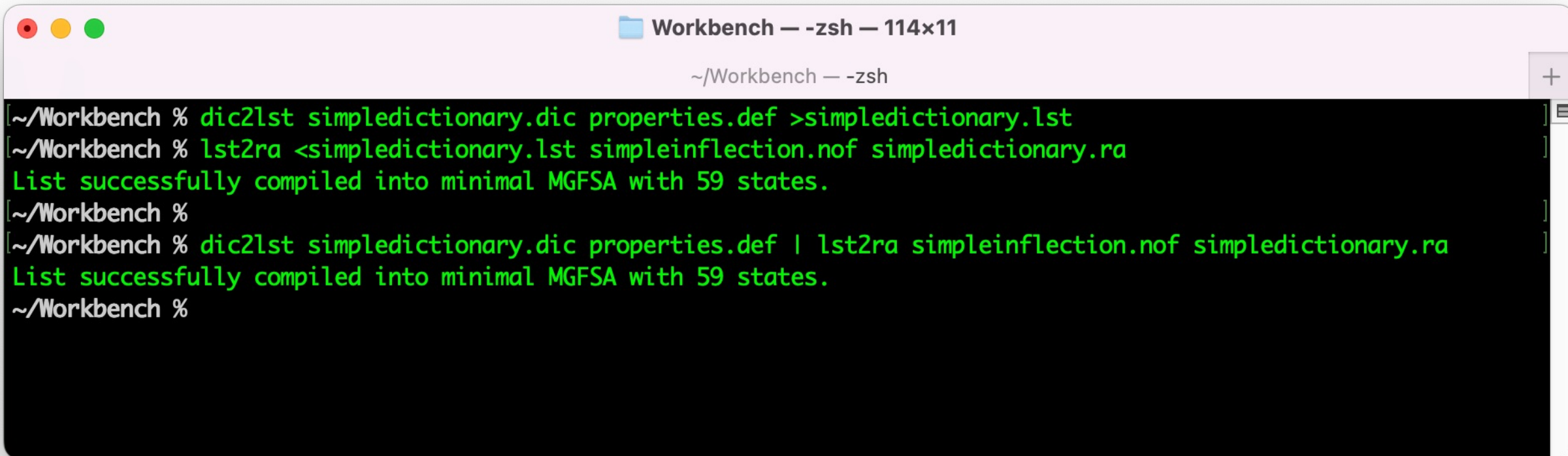
inputs:

- a list of all generated forms + inflectional and derivational grammars

outputs:

- a MGFSa

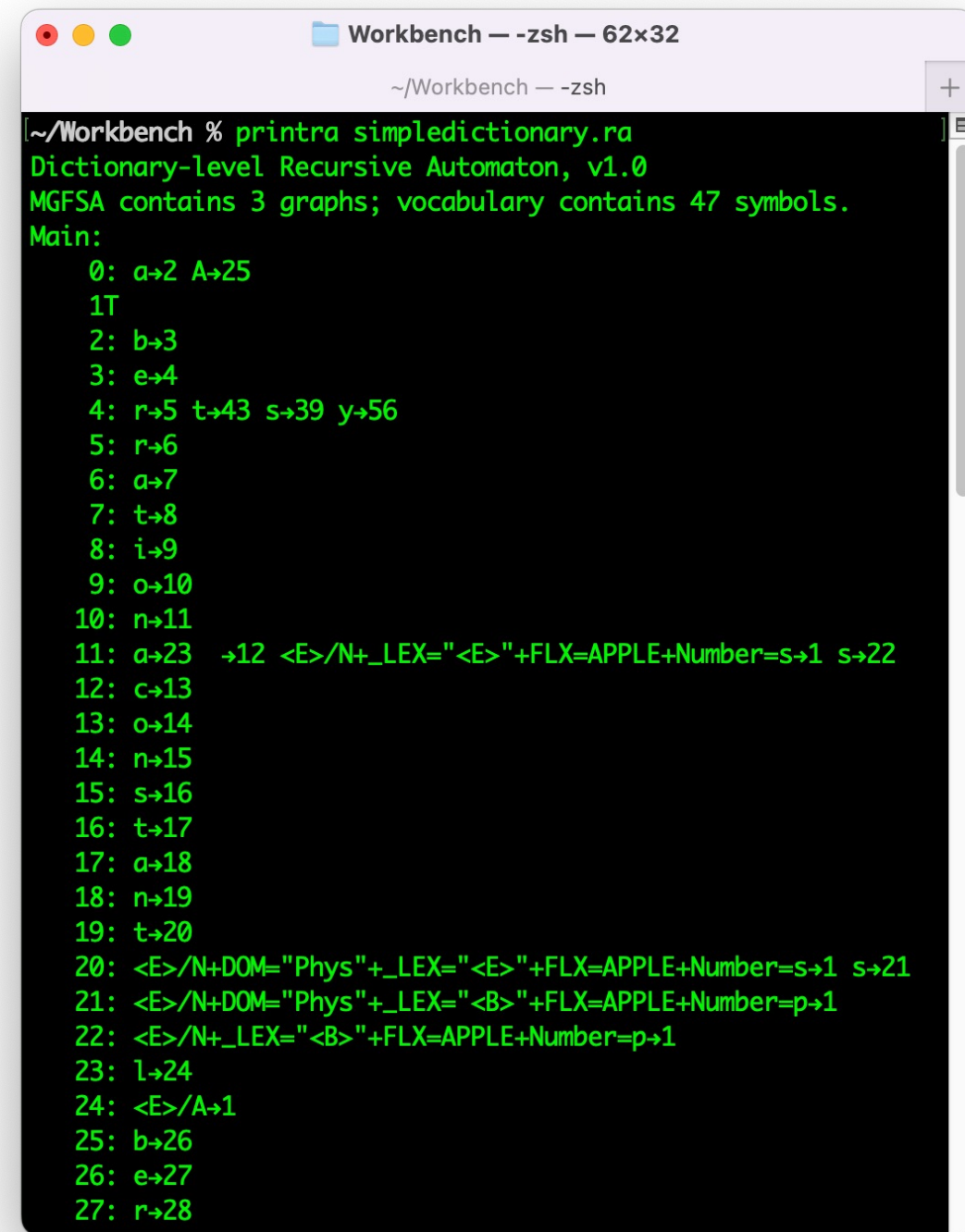
The MGFSa is reversible, i.e. can be used to parse or to generate texts



```
Workbench — -zsh — 114x11
~/Workbench — -zsh
[~/Workbench % dic2lst simplifiedictionary.dic properties.def >simplifiedictionary.lst
[~/Workbench % lst2ra <simplifiedictionary.lst simpleinflection.nof simplifiedictionary.ra
List successfully compiled into minimal MGFSa with 59 states.
[~/Workbench %
[~/Workbench % dic2lst simplifiedictionary.dic properties.def | lst2ra simpleinflection.nof simplifiedictionary.ra
List successfully compiled into minimal MGFSa with 59 states.
~/Workbench %
```

printra

prints a readable version of a MGFSFA
(.ra file)

A terminal window titled "Workbench — -zsh — 62x32" with a subtitle "~/Workbench — -zsh". The prompt is "[~/Workbench %]". The command "printra simplifiedictionary.ra" has been executed. The output is as follows:

```
Dictionary-level Recursive Automaton, v1.0
MGFSFA contains 3 graphs; vocabulary contains 47 symbols.
Main:
  0: a→2 A→25
  1T
  2: b→3
  3: e→4
  4: r→5 t→43 s→39 y→56
  5: r→6
  6: a→7
  7: t→8
  8: i→9
  9: o→10
 10: n→11
 11: a→23 →12 <E>/N+_LEX="<E>"+FLX=APPLE+Number=s→1 s→22
 12: c→13
 13: o→14
 14: n→15
 15: s→16
 16: t→17
 17: a→18
 18: n→19
 19: t→20
 20: <E>/N+DOM="Phys"+_LEX="<E>"+FLX=APPLE+Number=s→1 s→21
 21: <E>/N+DOM="Phys"+_LEX="<B>"+FLX=APPLE+Number=p→1
 22: <E>/N+_LEX="<B>"+FLX=APPLE+Number=p→1
 23: l→24
 24: <E>/A→1
 25: b→26
 26: e→27
 27: r→28
```

... (3 graphs; 47 symbol; 59 + 4 + 18 states).

nom2ra

input:

- a .nom graphical or textual grammar

output:

- the corresponding MGFSAs

options:

- determinize
- minimize

```
Workbench — -zsh — 62x38
~/Workbench — -zsh

~/Workbench % cat simpleregexp.nom
Main = (alb)*ab(alb)* ;
~/Workbench % nom2ra simpleregexp.nom
Orthographic grammar simpleregexp.ra successfully compiled.
1 orthographic grammar successfully compiled.
~/Workbench % printra simpleregexp.ra
Orthographic-level Recursive Automaton, v1.0
MGFSAs contains 1 graph; alphabet contains 3 symbols.
Main:
  0: <E>→2
  1T
  2: <E>→3 <E>→4
  3: <E>→10
  4: <E>→6 <E>→8
  5: <E>→2
  6: a→7
  7: <E>→5
  8: b→9
  9: <E>→5
 10: <E>→12
 11: <E>→1
 12: a→13
 13: <E>→14
 14: <E>→16
 15: <E>→11
 16: b→17
 17: <E>→18
 18: <E>→19 <E>→20
 19: <E>→15
 20: <E>→22 <E>→24
 21: <E>→18
 22: a→23
 23: <E>→21
 24: b→25
 25: <E>→21

~/Workbench %
```

```
Workbench — -zsh — 61x28
~/Workbench — -zsh

~/Workbench % nom2ra simpleregexp.nom -determinize
Orthographic grammar simpleregexp.ra successfully compiled.
1 orthographic grammar successfully compiled.
~/Workbench % printra simpleregexp.ra
Orthographic-level Recursive Automaton, v1.0
MGFSAs contains 1 graph; alphabet contains 3 symbols.
Main:
  0: a→1 b→2
  1: a→1 b→3
  2: a→1 b→2
 3T a→4 b→5
 4T a→4 b→6
 5T a→4 b→5
 6T a→4 b→5

~/Workbench % nom2ra simpleregexp.nom -minimize
Orthographic grammar simpleregexp.ra successfully compiled.
1 orthographic grammar successfully compiled.
~/Workbench % printra simpleregexp.ra
Orthographic-level Recursive Automaton, v1.0
MGFSAs contains 1 graph; alphabet contains 3 symbols.
Main:
  0: a→1 b→0
  1: a→1 b→2
 2T a→2 b→2

~/Workbench %
```


nog2ra

input:

- a .nog graphical or textual grammar

- the corresponding MGFSFA in the corresponding .ra file

options:

- determinize
- minimize

```
Workbench -- -zsh -- 58x37
~/Workbench -- -zsh
~/Workbench % cat np.nog
Main = <E>/<NP
      ( :PLURAL_DET (<A> | <E>) <N+p> |
        :SINGULAR_DET (<A> | <E>) <N+s> )
      <E>/> ;
PLURAL_DET = certain | few | many | the ;
SINGULAR_DET =
  a | an | some | the | this |
  part of the | such a | the flood of ;
~/Workbench % nog2ra np.nog -minimize
Syntactic grammar np.ra successfully compiled.
~/Workbench % printra np.ra
Syntactic-level Recursive Automaton, v1.0
MGFSFA contains 3 graphs; vocabulary contains 20 symbols.
Main:
  0: <E>/<NP→6
  1T
  2: <E>/>→1
  3: <N+p>→2
  4: <A>→5 <N+s>→2
  5: <N+s>→2
  6: :PLURAL_DET→7 :SINGULAR_DET→4
  7: <A>→3 <N+p>→2
PLURAL_DET:
  0: many→1 the→1 certain→1 few→1
  1T
SINGULAR_DET:
  0: the→2 a→1 an→1 some→1 this→1 part→6 such→5
  1T
  2T flood→4
  3: the→1
  4: of→1
  5: a→1
  6: of→3
~/Workbench %
```

ra2lst

input:
- an MGFSFA

output:
- the list of
generated
sequences

options:
- limit number of
loops

```
Workbench — -zsh — 73x22
~/Workbench — -zsh

~/Workbench % cat csar.nom
Main = (clt)/t (slz)/s ar/ar,N+Hum (<E>/+m l ina/+f) (<E>/+s l s/+p) ;
~/Workbench % ra2lst csar.ra
csar,tsar,N+Hum+m+s
csars,tsar,N+Hum+m+p
csarina,tsar,N+Hum+f+s
csarinas,tsar,N+Hum+f+p
czar,tsar,N+Hum+m+s
czars,tsar,N+Hum+m+p
czarina,tsar,N+Hum+f+s
czarinas,tsar,N+Hum+f+p
tsar,tsar,N+Hum+m+s
tsars,tsar,N+Hum+m+p
tsarina,tsar,N+Hum+f+s
tsarinas,tsar,N+Hum+f+p
tzar,tsar,N+Hum+m+s
tzars,tsar,N+Hum+m+p
tzarina,tsar,N+Hum+f+s
tzarinas,tsar,N+Hum+f+p
# 16 generated sequences.
~/Workbench %
```

```
Workbench — -zsh — 46x37
~/Workbench — -zsh

~/Workbench % cat np.nog
Main = <E>/<NP
      ( :PLURAL_DET (<A> l <E>) <N+p> l
        :SINGULAR_DET (<A> l <E>) <N+s> )
      <E>/> ;
PLURAL_DET = certain l few l many l the ;
SINGULAR_DET =
      a l an l some l the l this l
      part of the l such a l the flood of ;
~/Workbench % ra2lst np.ra
many <A> <N+p>,<NP>
many <N+p>,<NP>
the <A> <N+p>,<NP>
the <N+p>,<NP>
certain <A> <N+p>,<NP>
certain <N+p>,<NP>
few <A> <N+p>,<NP>
few <N+p>,<NP>
the <A> <N+s>,<NP>
the <N+s>,<NP>
the flood of <A> <N+s>,<NP>
the flood of <N+s>,<NP>
a <A> <N+s>,<NP>
a <N+s>,<NP>
an <A> <N+s>,<NP>
an <N+s>,<NP>
some <A> <N+s>,<NP>
some <N+s>,<NP>
this <A> <N+s>,<NP>
this <N+s>,<NP>
part of the <A> <N+s>,<NP>
part of the <N+s>,<NP>
such a <A> <N+s>,<NP>
such a <N+s>,<NP>
# 24 generated sequences.
~/Workbench %
```

ra2lst

input:

- an MGFSFA

output:

- the list of generated sequences

options:

- limit number of loops

```
Workbench — -zsh — 112x41
~/Workbench — -zsh

~/Workbench % cat simplifiedictionary2.dic
#use simpleinflection2.nof
the,DET
tsar,N+FLX=APPLE
mount,V+FLX=ASK
drink,N+FLX=APPLE
drink,V+FLX=DRINK
ultra,PREFIX
microscopic,PREFIX
tear,N+FLX=APPLE
~/Workbench % dic2lst simplifiedictionary2.dic properties.def | lst2ra simpleinflection2.nof simplifiedictionary2.ra
List successfully compiled into minimal MGFSFA with 49 states.
~/Workbench % ra2lst simplifiedictionary2.ra
drinking,V+_LEX="<B3>" +FLX=DRINK+Tense=G
drink,N+_LEX="<E>" +FLX=APPLE+Number=s
drinks,N+_LEX="<B>" +FLX=APPLE+Number=p
drinks,V+_LEX="<B>" +FLX=DRINK+Number=s+Pers=3+Tense=PR
drink,V+_LEX="<E>" +FLX=DRINK+Number=p+Pers=123+Tense=PR
drink,V+_LEX="<E>" +FLX=DRINK+Tense=INF
drink,V+_LEX="<E>" +FLX=DRINK+Number=s+Pers=12+Tense=PR
drank,V+_LEX="<L2><B>i<R2>" +FLX=DRINK+Number=s+Pers=123+Tense=PT
drank,V+_LEX="<L2><B>i<R2>" +FLX=DRINK+Number=p+Pers=123+Tense=PT
drunk,V+_LEX="<L2><B>i<R2>" +FLX=DRINK+Tense=PP
ultra,PREFIX
the,DET
tsar,N+_LEX="<E>" +FLX=APPLE+Number=s
tsars,N+_LEX="<B>" +FLX=APPLE+Number=p
tear,N+_LEX="<E>" +FLX=APPLE+Number=s
tears,N+_LEX="<B>" +FLX=APPLE+Number=p
microscopic,PREFIX
mounting,V+_LEX="<B3>" +FLX=ASK+Tense=G
mount,V+_LEX="<E>" +FLX=ASK+Number=p+Pers=123+Tense=PR
mount,V+_LEX="<E>" +FLX=ASK+Number=s+Pers=12+Tense=PR
mount,V+_LEX="<E>" +FLX=ASK+Tense=INF
mounts,V+_LEX="<B>" +FLX=ASK+Number=s+Pers=3+Tense=PR
mounted,V+_LEX="<B2>" +FLX=ASK+Tense=PP
mounted,V+_LEX="<B2>" +FLX=ASK+Number=s+Pers=123+Tense=PT
mounted,V+_LEX="<B2>" +FLX=ASK+Number=p+Pers=123+Tense=PT
# 25 generated sequences.
~/Workbench %
```


segment

inputs:

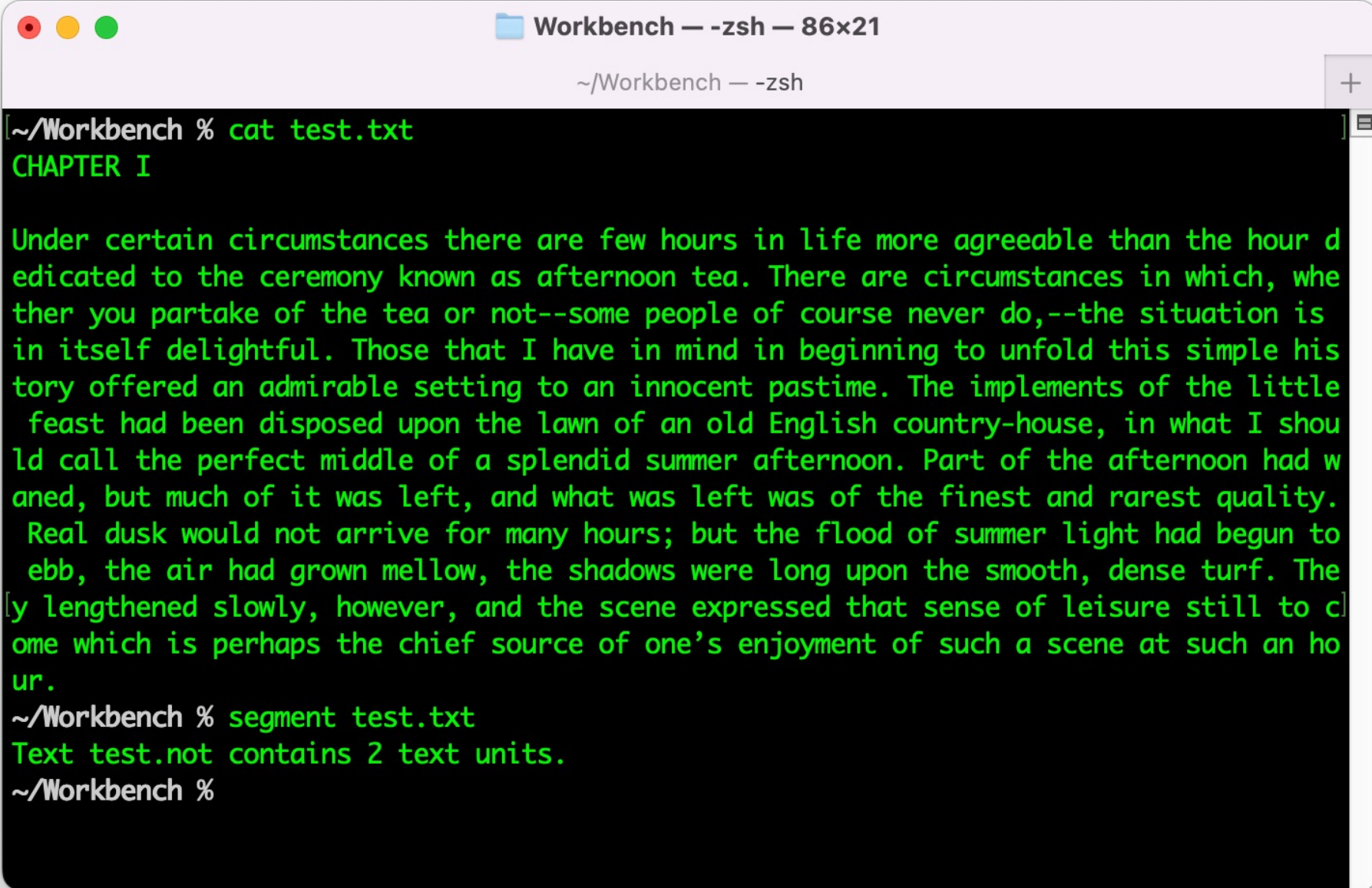
- texts in UTF8

outputs:

- annotated
segmented texts

option:

- a segmentation
orthographic
grammar



```
Workbench — -zsh — 86x21
~/Workbench — -zsh
[~/Workbench % cat test.txt
CHAPTER I

Under certain circumstances there are few hours in life more agreeable than the hour d
edicated to the ceremony known as afternoon tea. There are circumstances in which, whe
ther you partake of the tea or not--some people of course never do,--the situation is
in itself delightful. Those that I have in mind in beginning to unfold this simple his
tory offered an admirable setting to an innocent pastime. The implements of the little
feast had been disposed upon the lawn of an old English country-house, in what I shou
ld call the perfect middle of a splendid summer afternoon. Part of the afternoon had w
aned, but much of it was left, and what was left was of the finest and rarest quality.
Real dusk would not arrive for many hours; but the flood of summer light had begun to
ebb, the air had grown mellow, the shadows were long upon the smooth, dense turf. The
[y lengthened slowly, however, and the scene expressed that sense of leisure still to c
ome which is perhaps the chief source of one's enjoyment of such a scene at such an ho
ur.
~/Workbench % segment test.txt
Text test.not contains 2 text units.
~/Workbench %
```

When no MGFSA is given, segments text paragraph per paragraph

segment

inputs:

- texts in UTF8

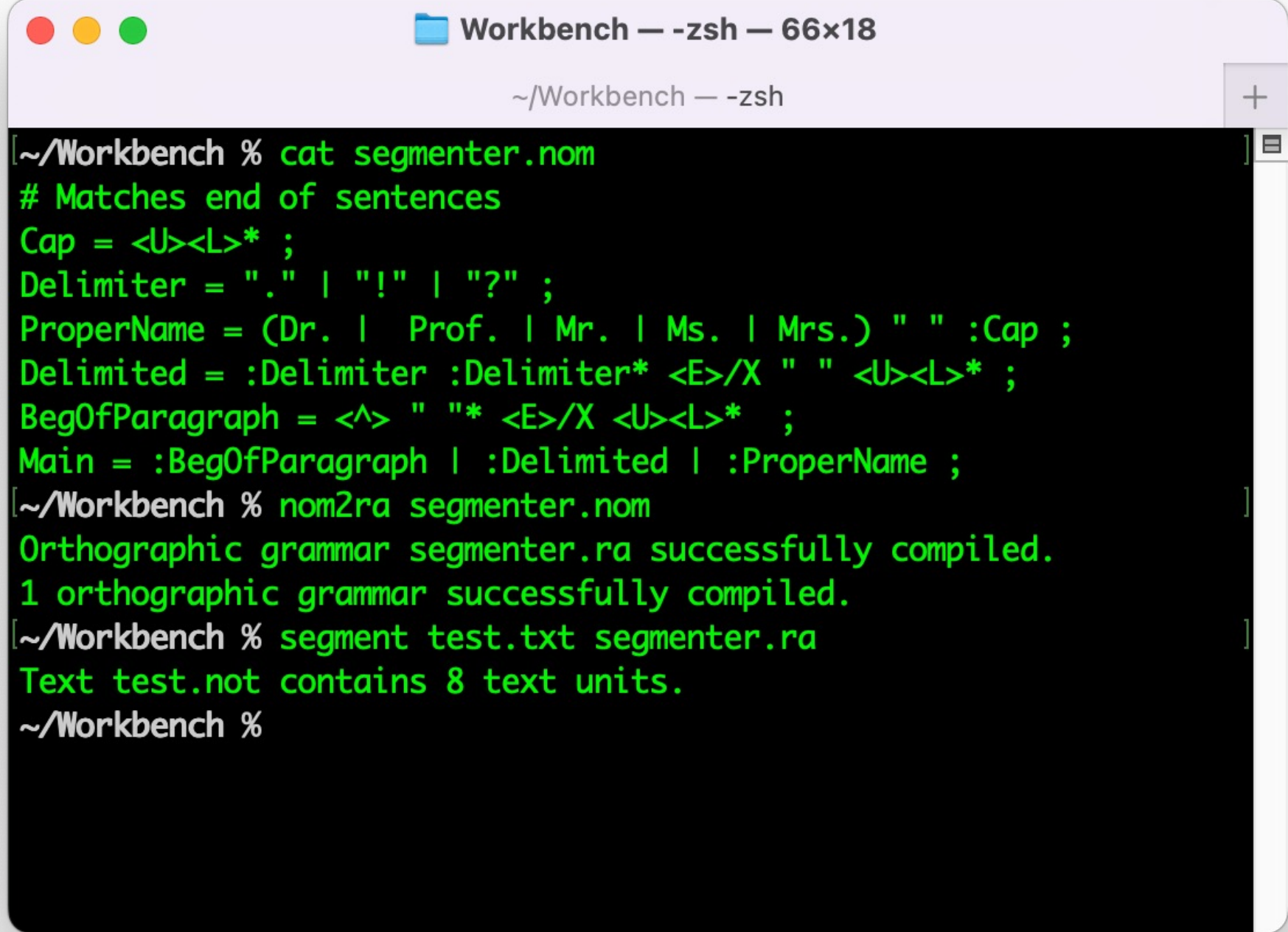
outputs:

- annotated

segmented texts

option:

- a segmentation orthographic grammar

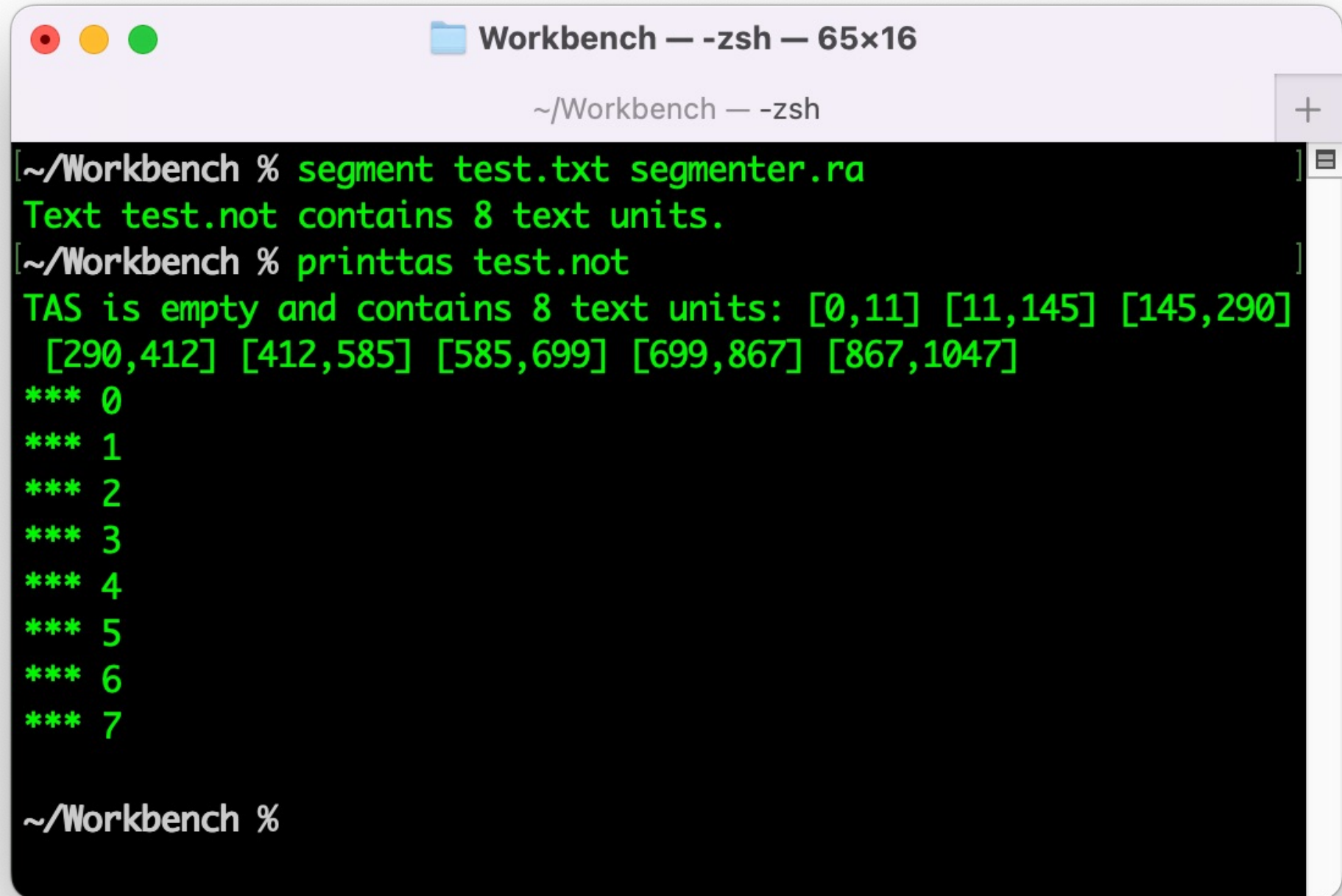


```
[~/Workbench % cat segmenter.nom
# Matches end of sentences
Cap = <U><L>* ;
Delimiter = "." | "!" | "?" ;
ProperName = (Dr. | Prof. | Mr. | Ms. | Mrs.) " " :Cap ;
Delimited = :Delimiter :Delimiter* <E>/X " " <U><L>* ;
BegOfParagraph = <^> " "* <E>/X <U><L>* ;
Main = :BegOfParagraph | :Delimited | :ProperName ;
[~/Workbench % nom2ra segmenter.nom
Orthographic grammar segmenter.ra successfully compiled.
1 orthographic grammar successfully compiled.
[~/Workbench % segment test.txt segmenter.ra
Text test.not contains 8 text units.
~/Workbench %
```

Applies a MGFSA to recognize text units (here: sentences)

printtas

prints a readable
version of a TAS



```
Workbench — -zsh — 65x16
~/Workbench — -zsh
[~/Workbench % segment test.txt segmenter.ra]
Text test.not contains 8 text units.
[~/Workbench % printtas test.not]
TAS is empty and contains 8 text units: [0,11] [11,145] [145,290]
[290,412] [412,585] [585,699] [699,867] [867,1047]
*** 0
*** 1
*** 2
*** 3
*** 4
*** 5
*** 6
*** 7

~/Workbench %
```


lexicalanalysis (1)

inputs:

- annotated texts
- MGFSAs
- a properties definitions file

outputs

- every text's TAS has been enriched

NOTE: only add
annotations at empty
positions

```
Workbench — -zsh — 97x25
~/Workbench — -zsh

~/Workbench % cat test2.txt
[The czar Zzzzist doesn't redismount drinkable ultramicroscopictears.%]
~/Workbench % segment test2.txt
Text test2.not contains 1 text unit.
~/Workbench % lexicalanalysis test2.not simplifiedictionary2.ra contractions.ra csar.ra multiplepref]
ixes.ra propernameism.ra reverb.ra verbable.ra properties.def
12 added annotations.
~/Workbench % printtas test2.not
TAS contains 12 annotations in 1 text unit: [0,68]
*** 0
000000 : <the,DET> → 000003
000004 : <tsar,N+Distribution=Hum+Gender=m+Number=s> → 000008
000009 : <Zzzz,N+Distribution=Hum+Number=s> → 000016
000017 : <do,V+Number=s+Pers=3+Tense=PR> → 000017.01
000017.01: <not,ADV> → 000024
000025 : <mount,V+Pref=dis+Pref=re+FLX=ASK+Number=p+Pers=123+Tense=PR> → 000035
000025 : <mount,V+Pref=dis+Pref=re+FLX=ASK+Number=s+Pers=12+Tense=PR> → 000035
000025 : <mount,V+Pref=dis+Pref=re+FLX=ASK+Tense=INF> → 000035
000036 : <drink,A+InitialCat=V+able> → 000045
000046 : <ultra,PREFIX> → 000046.01
000046.01: <microscopic,PREFIX> → 000046.02
000046.02: <tear,N+FLX=APPLE+Number=p> → 000067

~/Workbench %
```

lexicalanalysis (2)

inputs:

- annotated texts
- **a list of resources**
- a properties definitions file

outputs

- the TAS has been enriched

The image shows a code editor with two windows. The background window is titled 'listoflexicalresources.json' and shows a JSON array of objects. The foreground window is titled 'Workbench — -zsh — 84x22' and shows a terminal session.

```
~/Workbench/listoflexicalresources.json
[
  {
    "priority": 1,
    "path": "csar.ra"
  },
  {
    "priority": 1,
    "path": "contractions.ra"
  },
  {
    "priority": 1,
    "path": "..."
  },
  {
    "priority": 1,
    "path": "..."
  },
  {
    "priority": 1,
    "path": "..."
  },
  {
    "priority": 1,
    "path": "..."
  },
  {
    "priority": 1,
    "path": "..."
  },
  {
    "priority": 1,
    "path": "..."
  },
  {
    "priority": 1,
    "path": "..."
  },
  {
    "priority": 1,
    "path": "..."
  },
  {
    "priority": 1,
    "path": "..."
  },
  {
    "priority": 1,
    "path": "..."
  }
]
```

```
~/Workbench % segment test2.txt
Text test2.not contains 1 text unit.
~/Workbench % lexicalanalysis test2.not listoflexicalresources.json properties.def
12 added annotations.
~/Workbench % printtas test2.not
TAS contains 12 annotations in 1 text unit: [0,68]
*** 0
000000 : <the,DET> → 000003
000004 : <tsar,N+Distribution=Hum+Gender=m+Number=s> → 000008
000009 : <Zzzz,N+Distribution=Hum+Number=s> → 000016
000017 : <do,V+Number=s+Pers=3+Tense=PR> → 000017.01
000017.01: <not,ADV> → 000024
000025 : <mount,V+Pref=dis+Pref=re+FLX=ASK+Number=p+Pers=123+Tense=PR> → 000035
000025 : <mount,V+Pref=dis+Pref=re+FLX=ASK+Number=s+Pers=12+Tense=PR> → 000035
000025 : <mount,V+Pref=dis+Pref=re+FLX=ASK+Tense=INF> → 000035
000036 : <drink,A+InitialCat=V+able> → 000045
000046 : <ultra,PREFIX> → 000046.01
000046.01: <microscopic,PREFIX> → 000046.02
000046.02: <tear,N+FLX=APPLE+Number=p> → 000067

~/Workbench %
```


syntacticanalysis

inputs:

- annotated texts
- syntactic MGFSAs
- properties definitions file

output

- add annotations to the TAS

```
Workbench — -zsh — 101x36
~/Workbench — -zsh

~/Workbench % cat test3.txt
Under certain circumstances there are few hours in life more agreeable than the hour dedicated to the
ceremony known as afternoon tea.
~/Workbench % segment test3.txt ; lexicalanalysis test3.not simpledictionary3.ra properties.def
Text test3.not contains 1 text unit.
46 added annotations.
~/Workbench % cat np.nog
Main = <E>/<NP
      ( :PLURAL_DET (<A> | <E>) <N+p> |
        :SINGULAR_DET (<A> | <E>) <N+s> )
      <E>/> ;
PLURAL_DET = certain | few | many | the ;
SINGULAR_DET =
  a | an | some | the | this |
  part of the | such a | the flood of ;
~/Workbench % syntacticanalysis test3.not np.ra properties.def
4 added annotations; 0 removed annotation.
~/Workbench % printtas test3.not
TAS contains 50 annotations in 1 text unit: [0,134]
*** 0
000000 : <under,A+Human> → 000005
000000 : <under,ADV> → 000005
000000 : <under,PREFIX> → 000005
000000 : <under,PREP> → 000005
000006 : <NP> → 000027
000006 : <certain,A> → 000013
000006 : <certain,DET> → 000013
000014 : <circumstance,N+Distribution=Abst+FLX=APPLE+Number=p> → 000027
000028 : <there,ADV> → 000033
000028 : <there,INTJ> → 000033
000028 : <there,PRO+Number=p+Pers=3> → 000033
000028 : <there,PRO+Number=s+Pers=3> → 000033
000034 : <be,V+Auxiliary+FLX=BE+Number=p+Pers=123+Tense=PR> → 000037
000034 : <be,V+Auxiliary+FLX=BE+Number=s+Pers=2+Tense=PR> → 000037
000038 : <NP> → 000047
000038 : <few,DET+Number=p> → 000041
```

syntacticanalysis

inputs:

- annotated texts
- syntactic MGFSAs
- properties definitions file

output

- remove annotations from the TAS

```
Workbench — -zsh — 105x42
~/Workbench — -zsh

~/Workbench % cat test2.txt
The czar Zzzzist doesn't redismount drinkable ultramicroscopictears.

~/Workbench % segment test2.txt ; lexicalanalysis test2.not listoflexicalresources.json properties.def
Text test2.not contains 1 text unit.
12 added annotations.

~/Workbench % printtas test2.not
TAS contains 12 annotations in 1 text unit: [0,68]
*** 0
000000 : <the,DET> → 000003
000004 : <tsar,N+Distribution=Hum+Gender=m+Number=s> → 000008
000009 : <Zzzz,N+Distribution=Hum+Number=s> → 000016
000017 : <do,V+Number=s+Pers=3+Tense=PR> → 000017.01
000017.01: <not,ADV> → 000024
000025 : <mount,V+Pref=dis+Pref=re+FLX=ASK+Number=p+Pers=123+Tense=PR> → 000035
000025 : <mount,V+Pref=dis+Pref=re+FLX=ASK+Number=s+Pers=12+Tense=PR> → 000035
000025 : <mount,V+Pref=dis+Pref=re+FLX=ASK+Tense=INF> → 000035
000036 : <drink,A+InitialCat=V+able> → 000045
000046 : <ultra,PREFIX> → 000046.01
000046.01: <microscopic,PREFIX> → 000046.02
000046.02: <tear,N+FLX=APPLE+Number=p> → 000067

~/Workbench % cat disamb.nog
Main = <do> <ADV> <V>/<V+INF> ;

~/Workbench % nog2ra disamb.nog -minimize ; syntacticanalysis test2.not disamb.ra properties.def
Syntactic grammar disamb.ra successfully compiled.
0 added annotations. 2 removed annotations.

~/Workbench % printtas test2.not
TAS contains 10 annotations in 1 text unit: [0,68]
*** 0
000000 : <the,DET> → 000003
000004 : <tsar,N+Distribution=Hum+Gender=m+Number=s> → 000008
000009 : <Zzzz,N+Distribution=Hum+Number=s> → 000016
000017 : <do,V+Number=s+Pers=3+Tense=PR> → 000017.01
000017.01: <not,ADV> → 000024
000025 : <mount,V+Pref=dis+Pref=re+FLX=ASK+Tense=INF> → 000035
000036 : <drink,A+InitialCat=V+able> → 000045
000046 : <ultra,PREFIX> → 000046.01
000046.01: <microscopic,PREFIX> → 000046.02
000046.02: <tear,N+FLX=APPLE+Number=p> → 000067

~/Workbench %
```

syntacticlocate

inputs:

- a list of texts
- one syntactic query'

MGFSA

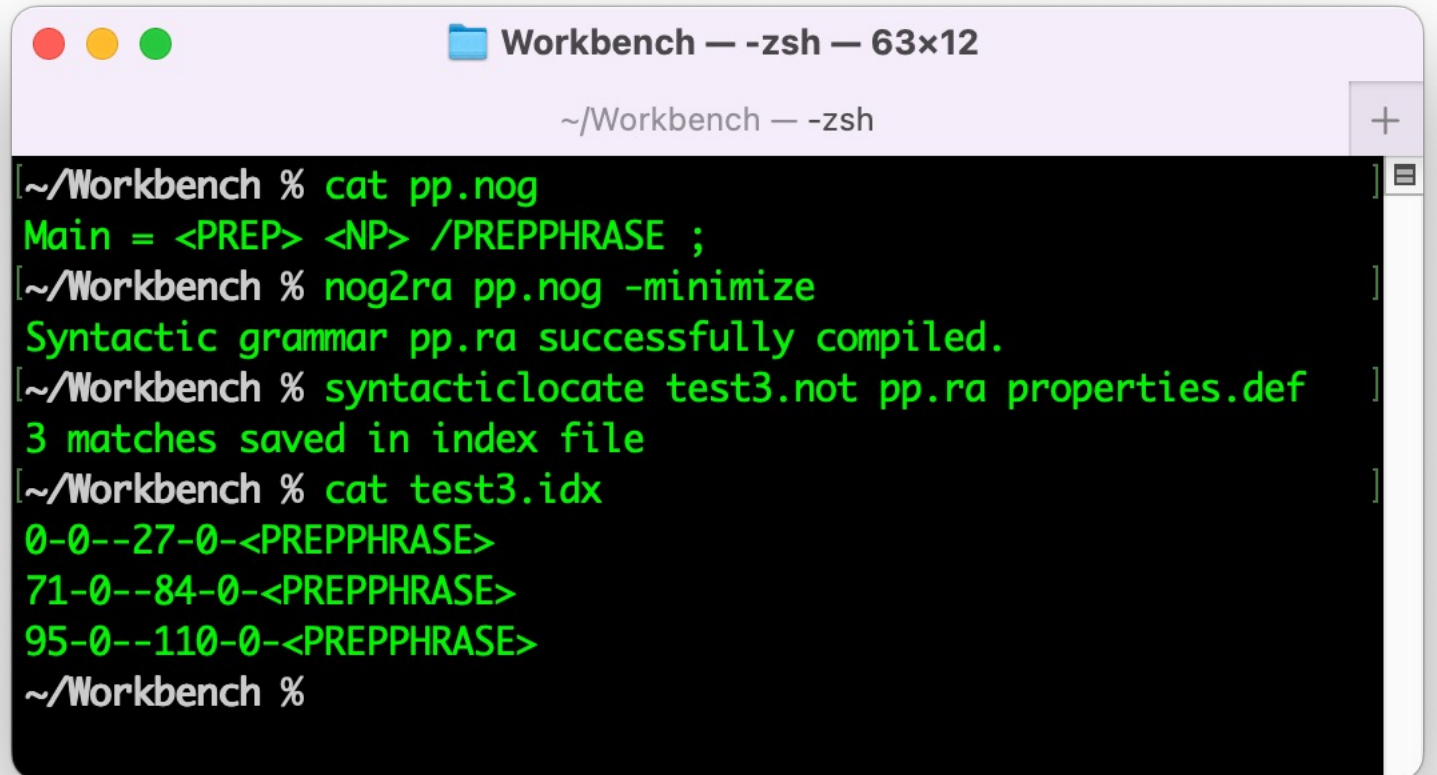
- a properties definitions file

option:

- shortest, longest, all

outputs:

- the index files that contains the matches in every text



```
Workbench — -zsh — 63x12
~/Workbench — -zsh
[~/Workbench % cat pp.nog
Main = <PREP> <NP> /PREPPHRASE ;
[~/Workbench % nog2ra pp.nog -minimize
Syntactic grammar pp.ra successfully compiled.
[~/Workbench % syntacticlocate test3.not pp.ra properties.def
3 matches saved in index file
[~/Workbench % cat test3.idx
0-0--27-0-<PREPPHRASE>
71-0--84-0-<PREPPHRASE>
95-0--110-0-<PREPPHRASE>
~/Workbench %
```

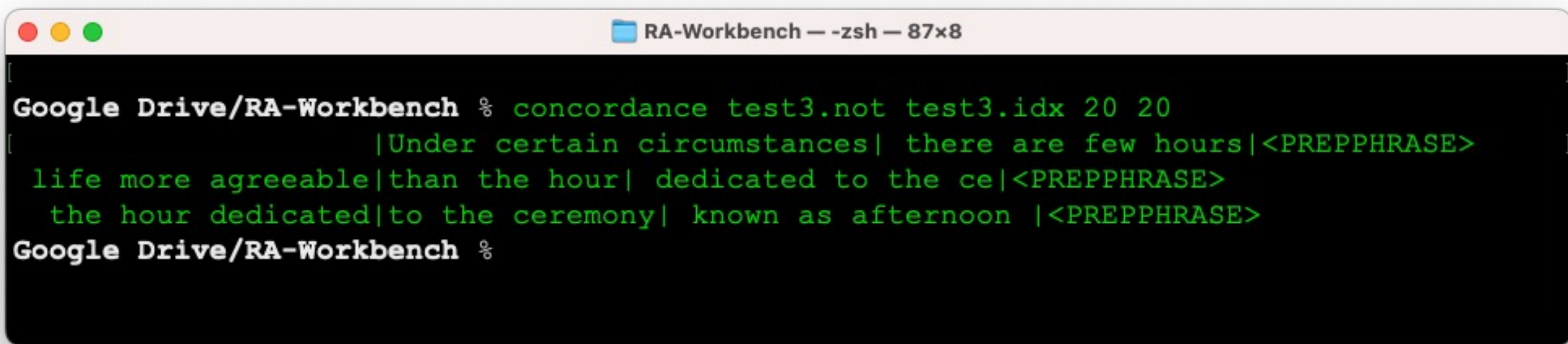

concordance

parameters:

- a text, e.g.: test.not
- a list of indices, e.g.: test1.idx test2.idx test3.idx
- left and right columns' lengths

output:

- the concordance



```
RA-Workbench — -zsh — 87x8

Google Drive/RA-Workbench % concordance test3.not test3.idx 20 20
[
    |Under certain circumstances| there are few hours|<PREPPHRASE>
life more agreeable|than the hour| dedicated to the ce|<PREPPHRASE>
the hour dedicated|to the ceremony| known as afternoon |<PREPPHRASE>
Google Drive/RA-Workbench %
```

Perspective

- Aiming at v1.0: please send **feedback**!
- Formalize discontinuous expressions (e.g. phrasal verbs) in a easier way than NooJ's
- Implement a better transformational engine than NooJ's
- Adapt NooJ resources for the 30+ supported languages
- Add a graphical interface (WEB?)
- Add ATISHS' statistical functionalities, see <http://atish.univ-fcomte.fr>
- Collaborative... **Collaborations**, anyone?