Software Languages

language - languages - software

Guido Wachsmuth, Eelco Visser


BIGGER PICTURE

language

definition

properties

study

languages

natural

controlled

artificial

software

language software

language processors

compilers

compiler construction

language definition

Language is the source of misunderstandings.

Antoine de Saint-Exupéry: Le Petit Prince.

Language is

a purely human and non-instinctive method

of communicating ideas, emotions, and desires

by means

of a system of voluntarily produced symbols.

Edward Sapir: Language. An Introduction to the Study of Speech.

Language is

a **method**

of **communicating**

by means

of **symbols**.

Edward Sapir: Language. An Introduction to the Study of Speech.

Language is

a **purely human** and **non-instinctive method**

of **communicating**

by means

of a **system** of **voluntarily produced symbols**.

Edward Sapir: Language. An Introduction to the Study of Speech.

Language is

a **purely human** and **non-instinctive method**

of **communicating ideas**, **emotions**, and **desires**

by means

of a **system** of **voluntarily produced symbols**.

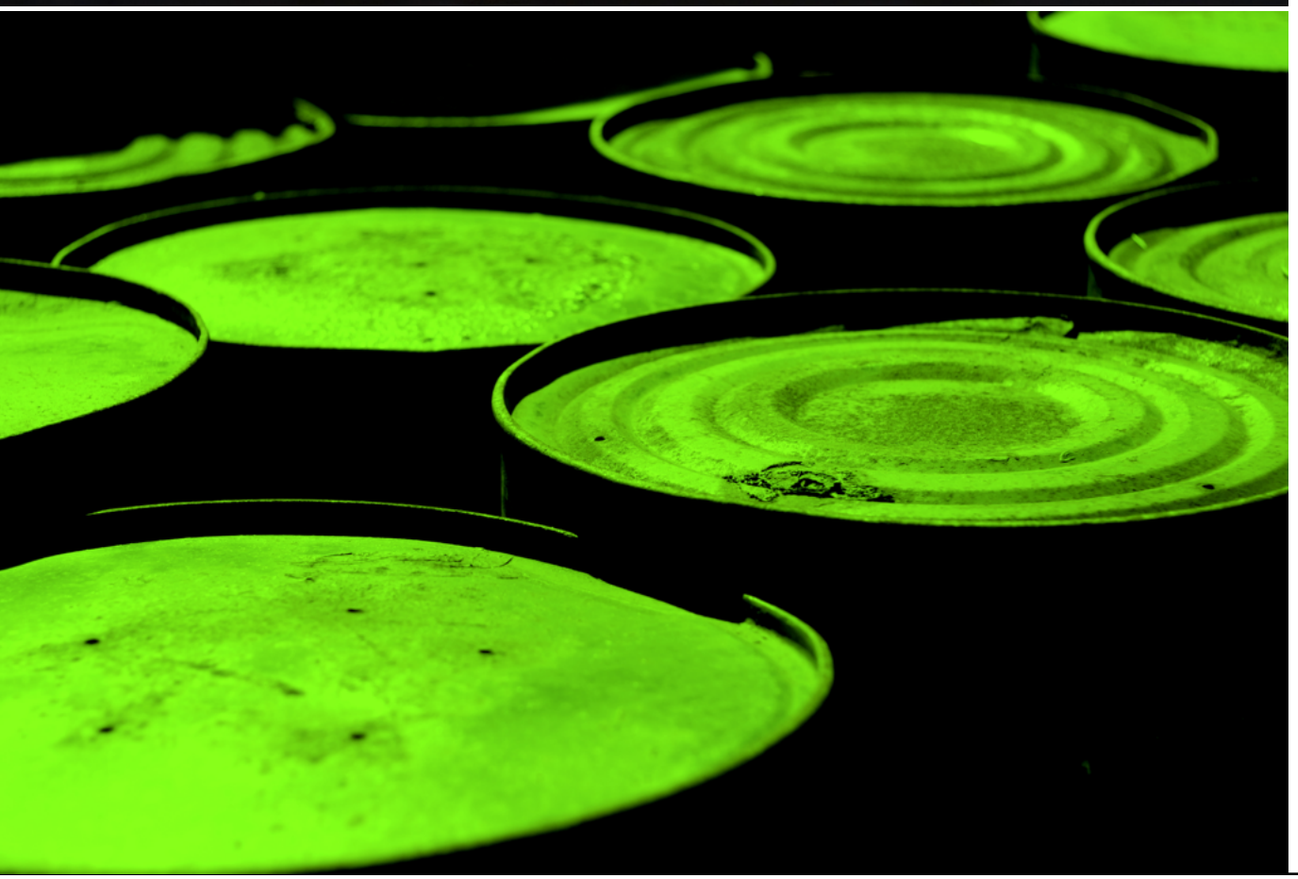
Edward Sapir: Language. An Introduction to the Study of Speech.

language properties

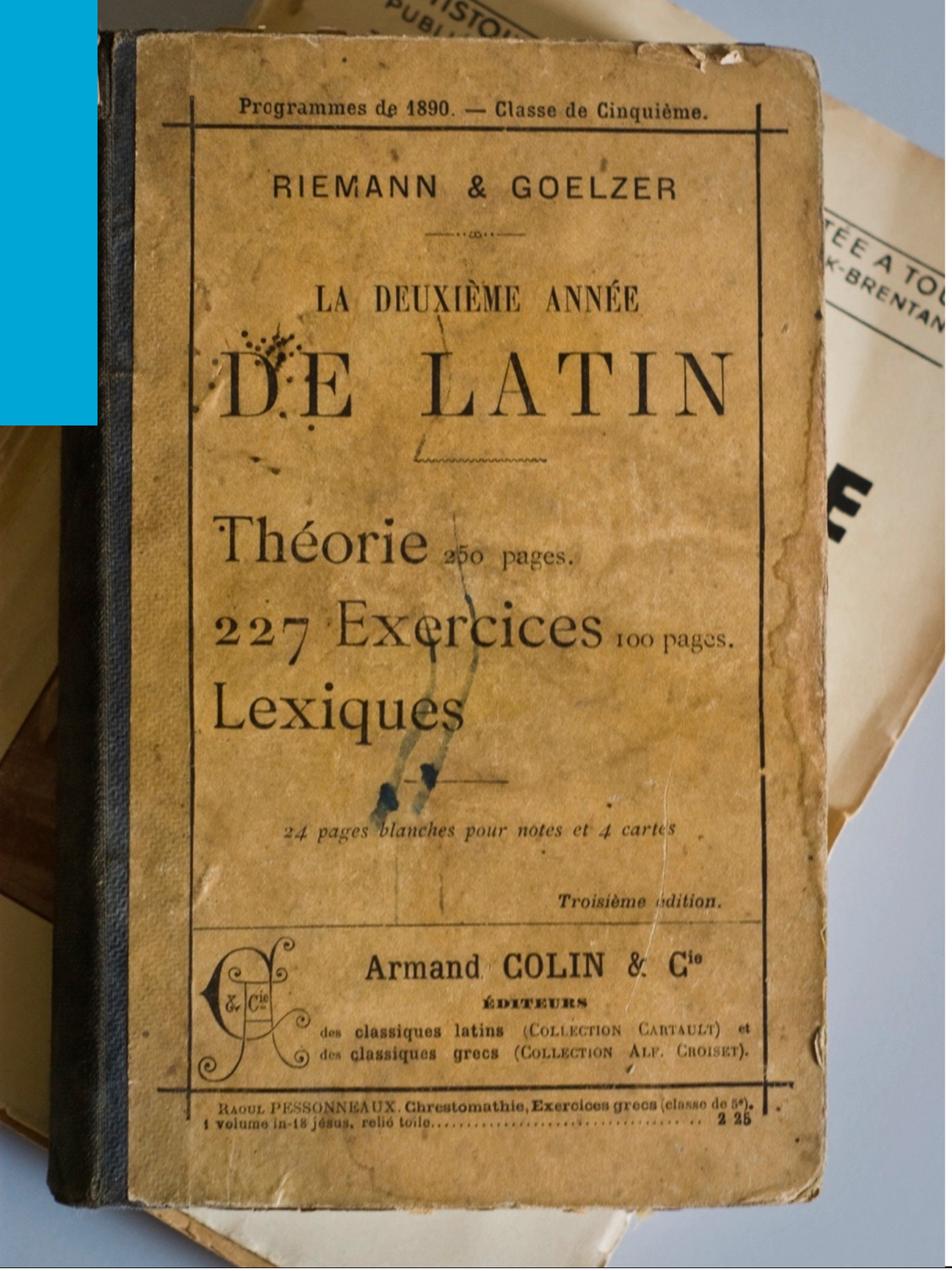
non-instinctive


Shoes
Refreshiug
Bear
Cross
child
Snow day
Wolf
Ring
Cricket
Song
Ship
Looking
moon
Cow
Box
Rat
Cuting
rice
Star
Fishing
Hoise
Penmanship
Cat
Blank book
Chesnut
WILD HOG
Horse car
Pig
Commandant
monkey
School
Creek
Gaot
Tea Po
Physician
PaRroT
Deer
Snake
Ba th
Crab
Dog
Bee
Chicken

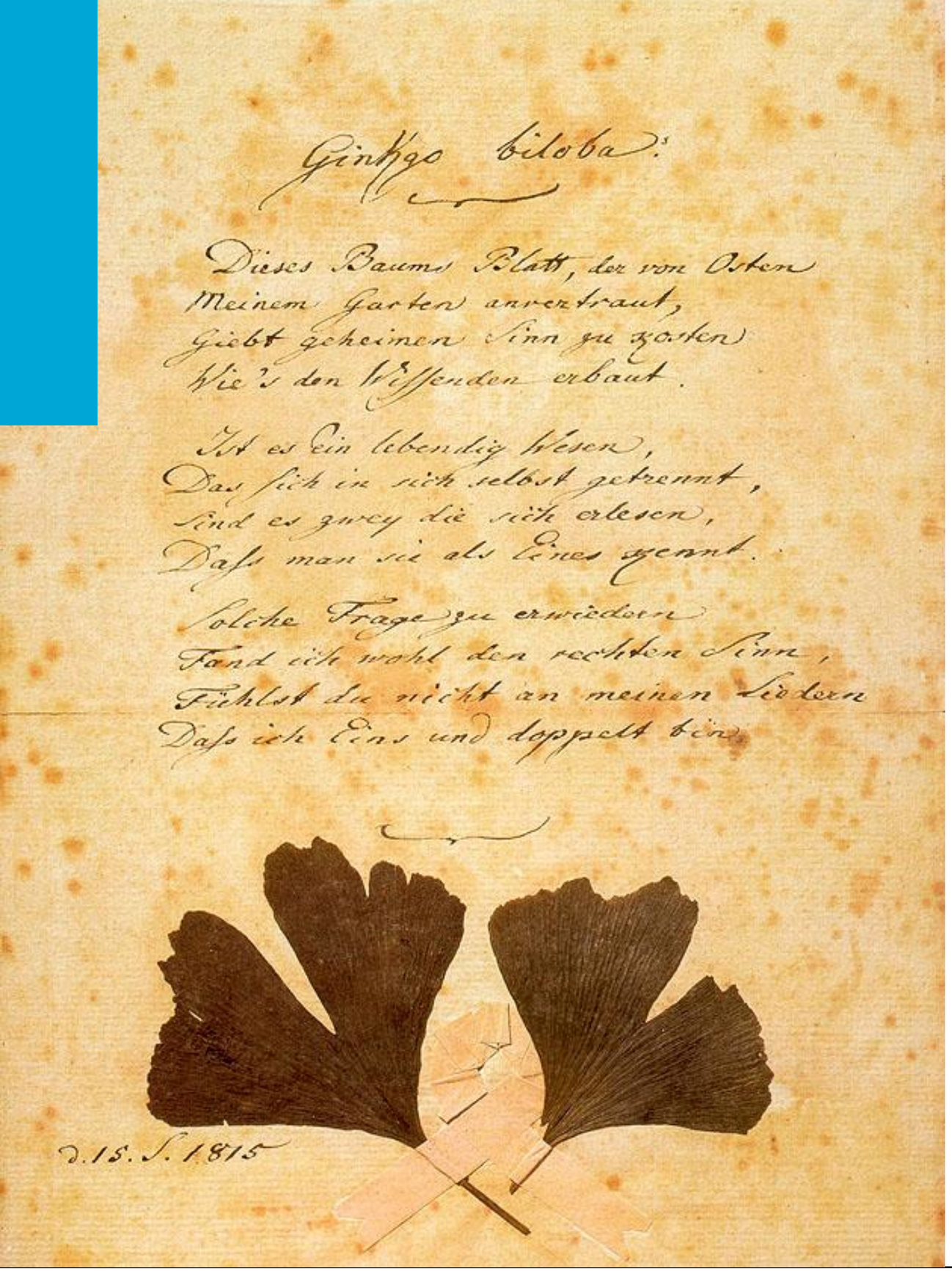
symbolic



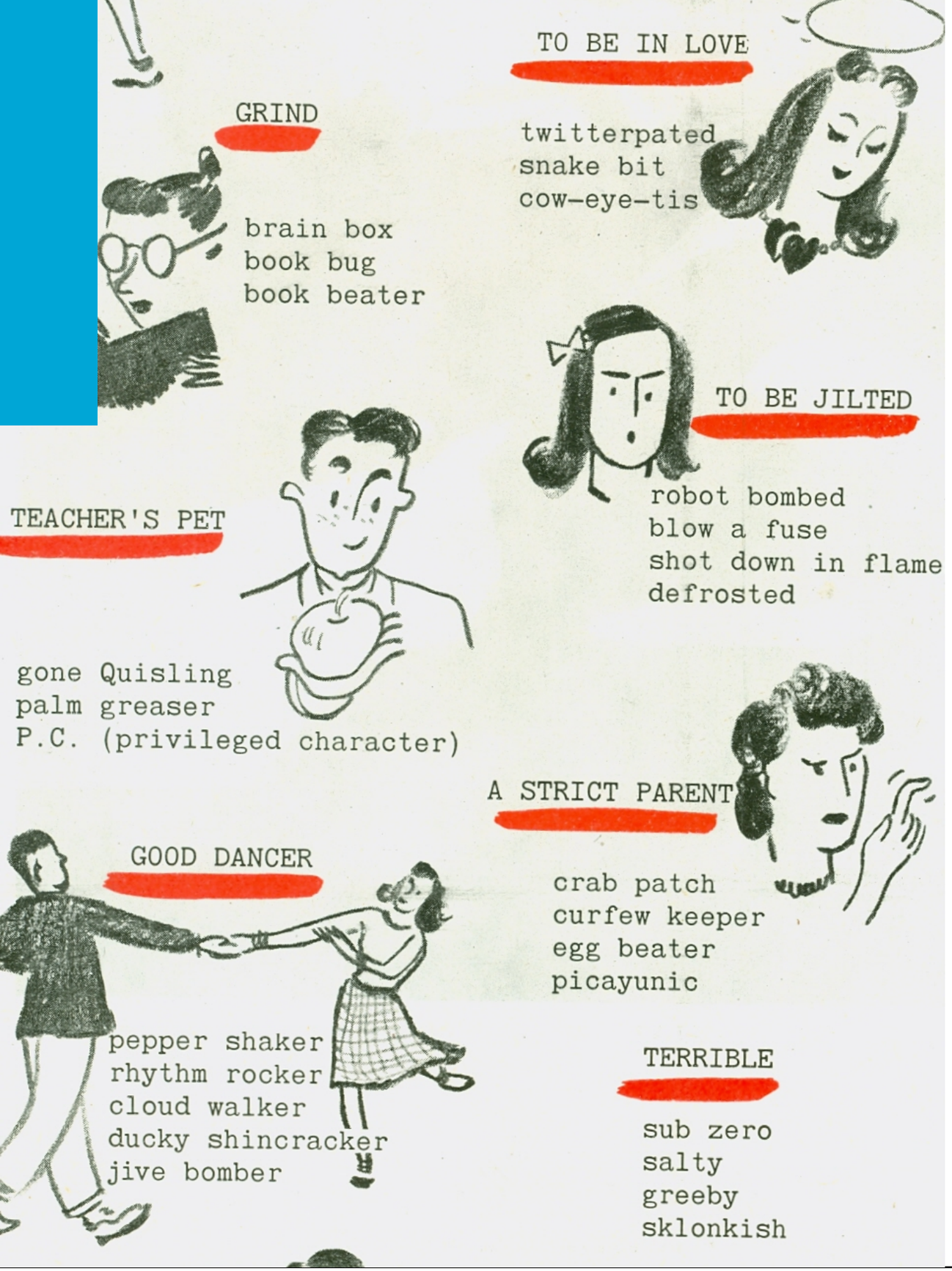
arbitrary


Programmes de 1890. — Classe de Cinquième.
RIEMANN & GOELZER
LA DEUXIÈME ANNÉE
DE LATIN
Théorie 250 pages.
227 Exercices 100 pages.
Lexiques
24 pages blanches pour notes et 4 cartes
Troisième édition.
Armand COLIN & Cie
ÉDITEURS
des classiques latins (COLLECTION CARTAULT) et
des classiques grecs (COLLECTION ALF. CROISET).
RAOUL PESSONNEAUX. Chrestomathie, Exercices grecs (classe de 5e).
1 volume in-18 jésus, relié toile..... 2 25

systematic


Ginkgo biloba.
Dieses Baums Blatt, der von Osten
Meinem Garten anvertraut,
Giebt geheimen Sinn zu kosten
Wie's den Wissenden erbaut.
Ist es ein lebendig Wesen,
Das sich in sich selbst getrennt,
Sind es zwey die sich erlesen,
Dass man sie als Eines kennt.
Solche Frage zu erwiedern
Fand ich wohl den rechten Sinn,
Fühlst du nicht an meinen Liedern
Dass ich Eins und doppelt bin.
d. 15. S. 1815

productive



conventional


xiii. corr. to OFris. wēnsdei, app
form with mutation (*wōdinaz) repl.
wōnsdei, MLG.
Mercurii diēs 'day of
OS. Wōden
Odin', tr. late L.
Mercury', OE. Wōden
tan, ON. Óðinn is referred
excited or inspired.

modifiable

language study


meta language facility

Philosophy

Linguistics

lexicology

grammar

morphology

syntax

phonology

semantics

Interdisciplinary

applied linguistics

computational linguistics

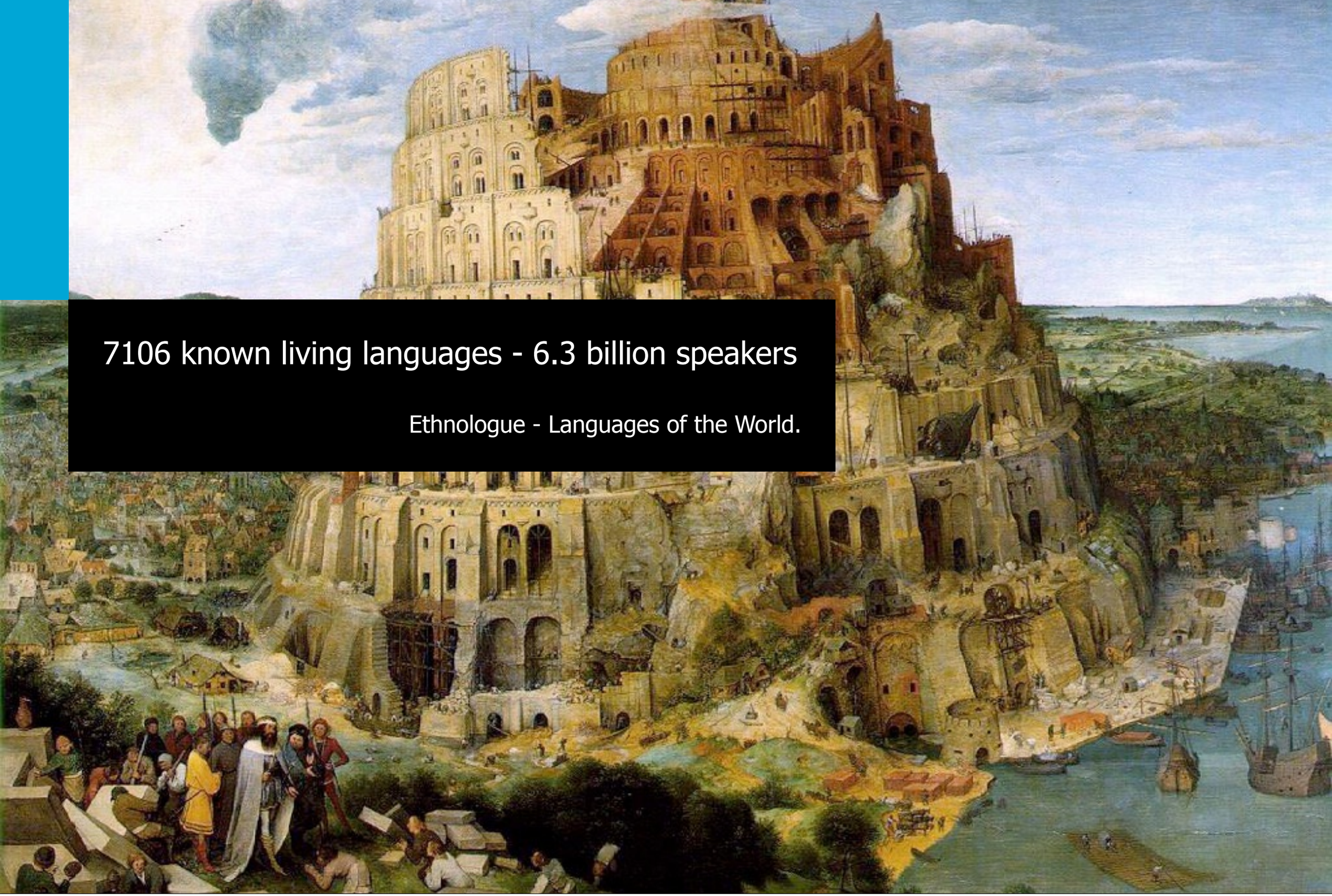
historical linguistics

neurolinguistics

psycholinguistics

sociolinguistics

natural languages



7106 known living languages - 6.3 billion speakers

Ethnologue - Languages of the World.

A language is a dialect with a navy and an army.

Max Weinreich: YVO Bletter 25(1).

controlled languages


meta language facility

BOOK VI.

THE ASHTÁDHYÁYÍ OF PÁṆINI.

TRANSLATED INTO ENGLISH

BY

SRISA CHANDRA VASU, B. A.,

Provincial Civil Service, N. W. P.

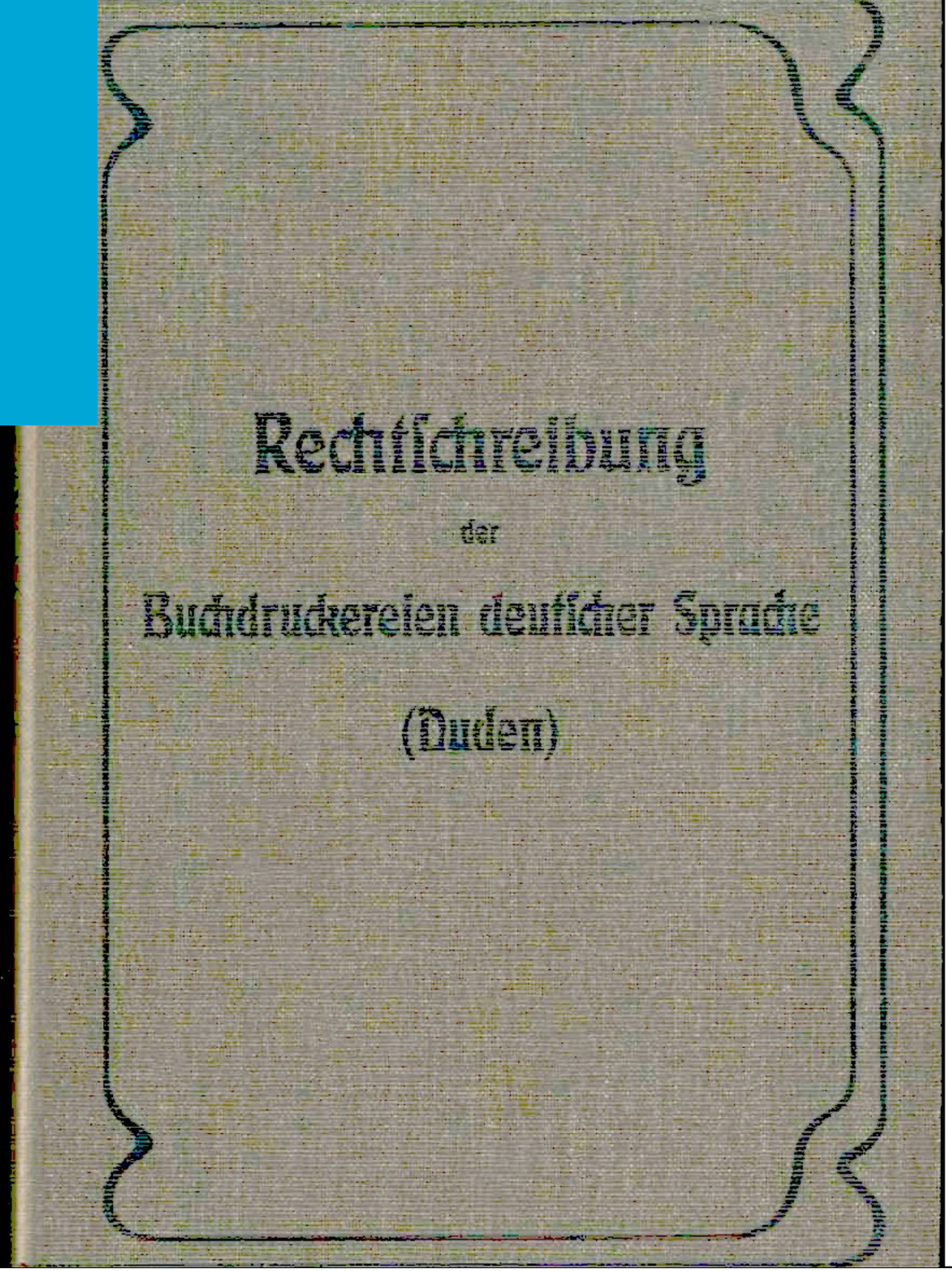
Benares:

PUBLISHED BY SINDHU CHARAN BOSE,

at the Panini Office,

1897.

Classical Sanskrit


Rechtschreibung
der
Buchdruckereien deutscher Sprache
(Duden)

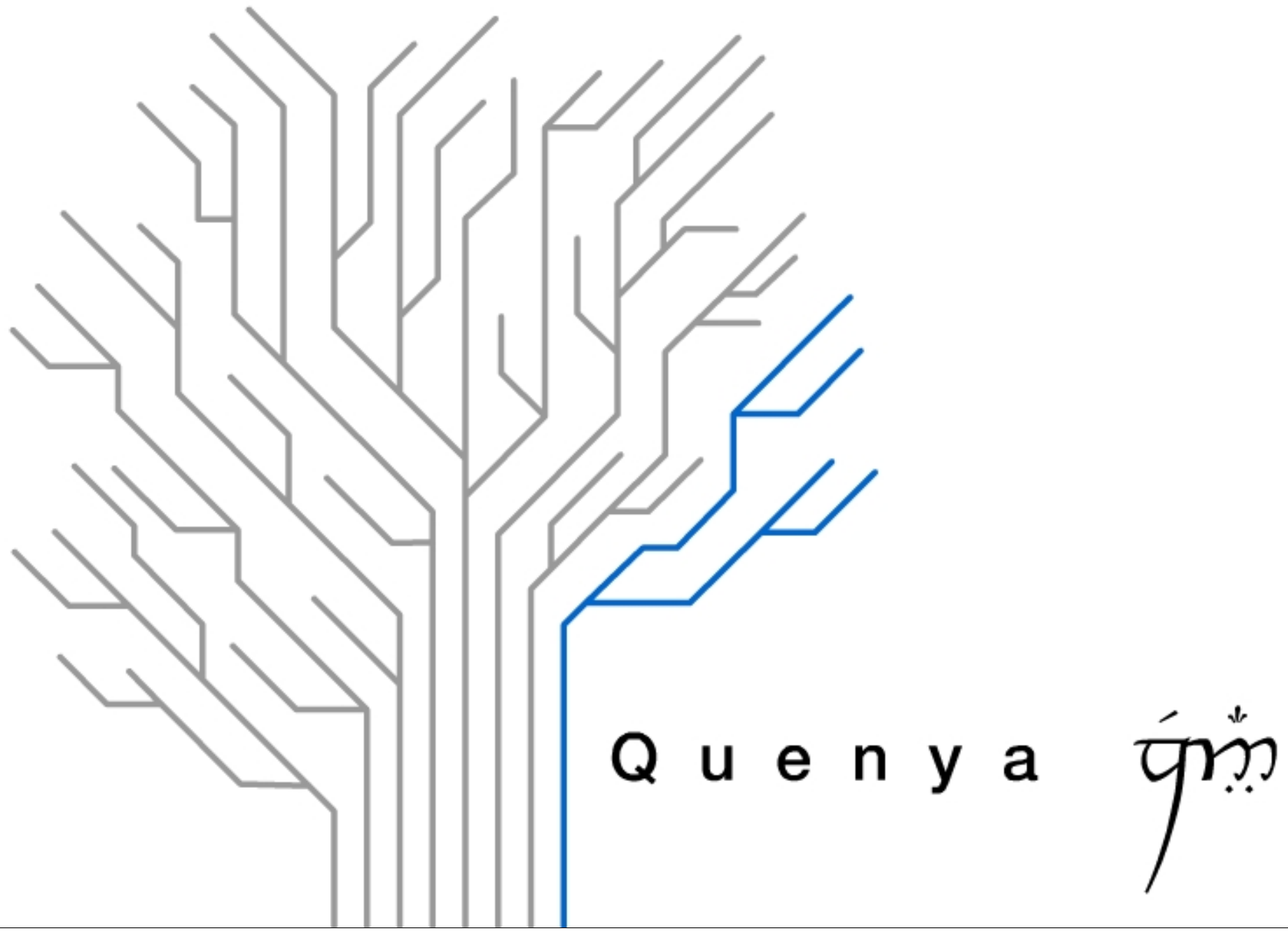
German

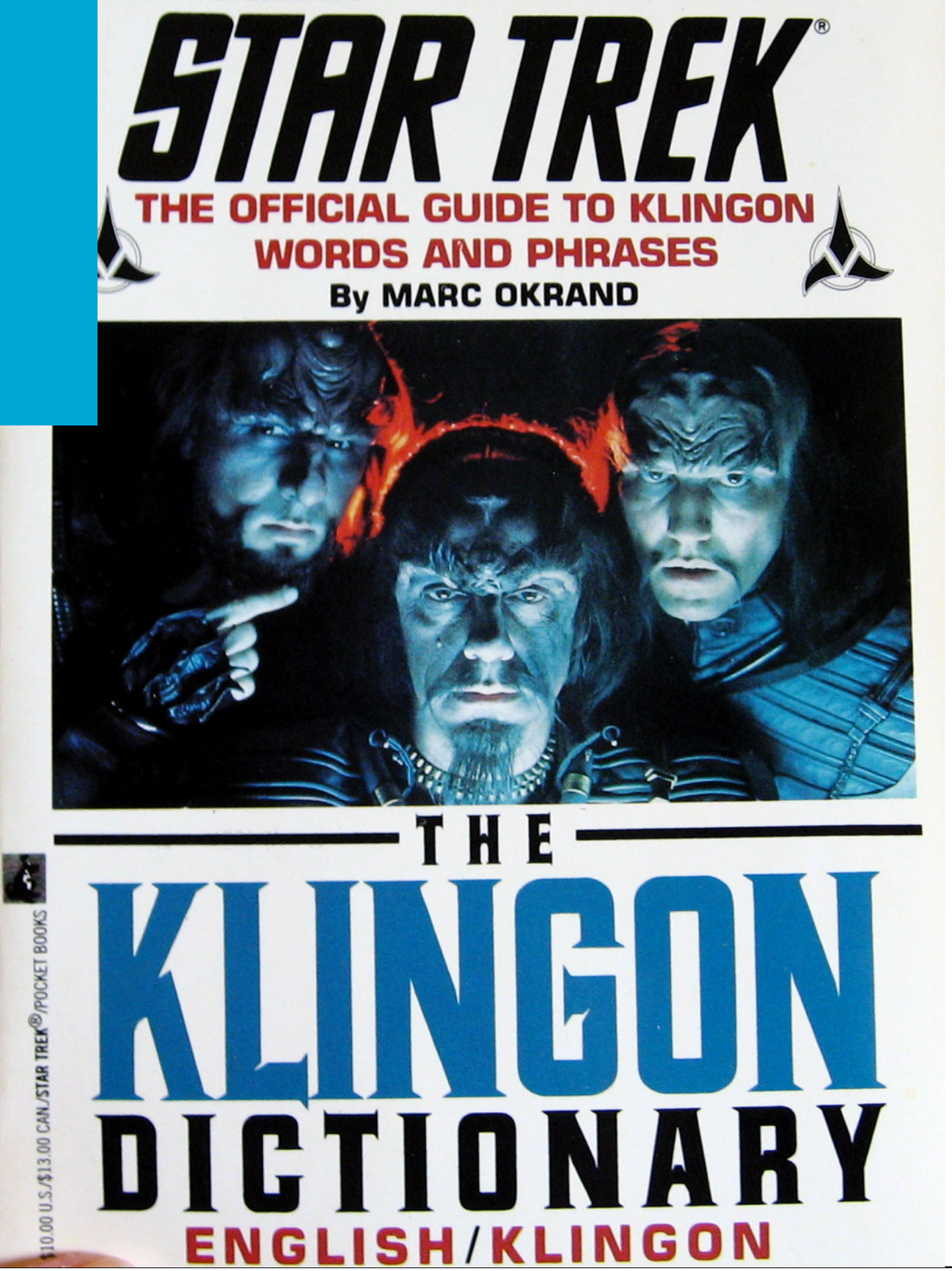
ASD Simplified

Technical English

artificial languages


ESPERANTO


Quenya


STAR TREK®
THE OFFICIAL GUIDE TO KLINGON WORDS AND PHRASES
By MARC OKRAND
THE KLINGON DICTIONARY
ENGLISH/KLINGON
$10.00 U.S./$13.00 CAN./STAR TREK®/POCKET BOOKS

Klingon

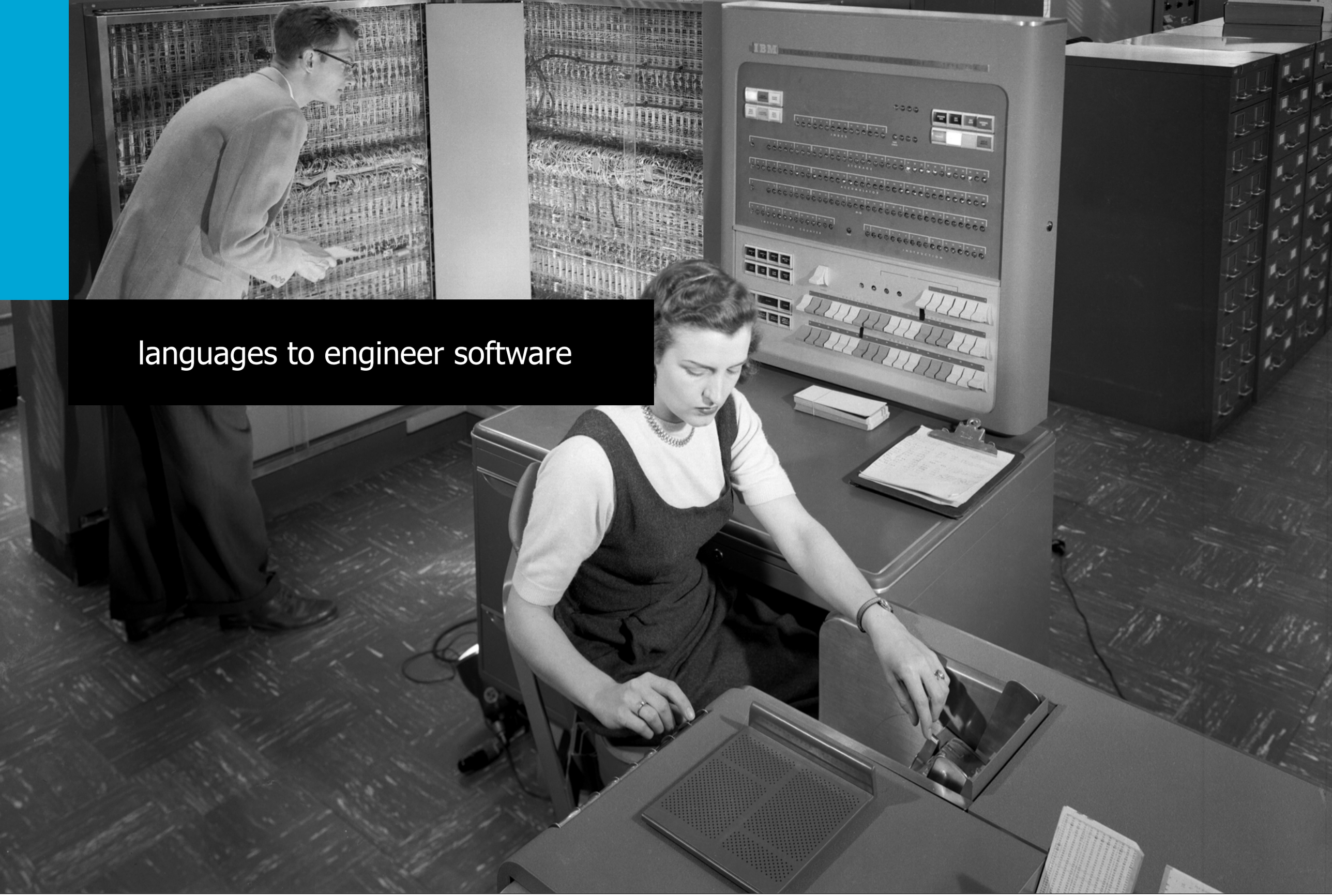
Na'vi

software languages

8512 programming languages - 18.5 million developers

Diarmuid Pigott: History of Programming Languages.
IDC: 2014 Worldwide Software Developer and ICT-Skilled Worker Estimates.

languages to engineer software



to provide a means

of communicating

numerical methods and other procedures

between people

John W. Backus: The Syntax and Semantics of the Proposed International Algebraic Language of Zürich ACM-GAMM Conference.

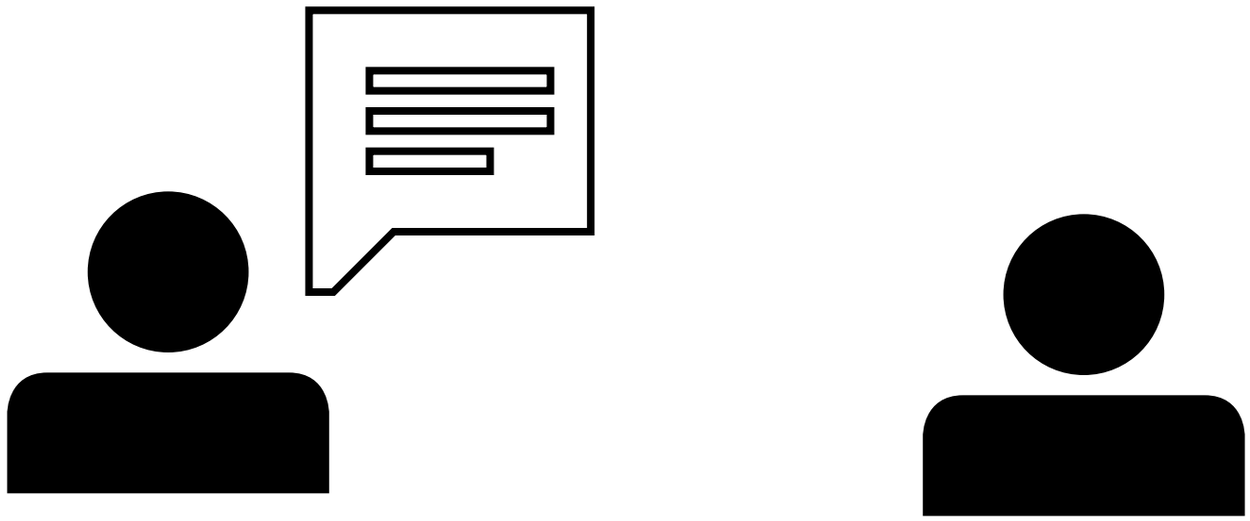
to provide a means

of **communicating**

numerical methods and **other procedures**

between **people**

John W. Backus: The Syntax and Semantics of the Proposed International Algebraic Language of Zürich ACM-GAMM Conference.



natural language

software language

Philosophy

Linguistics

lexicology

grammar

morphology

syntax

phonology

semantics

Interdisciplinary

Computer Science

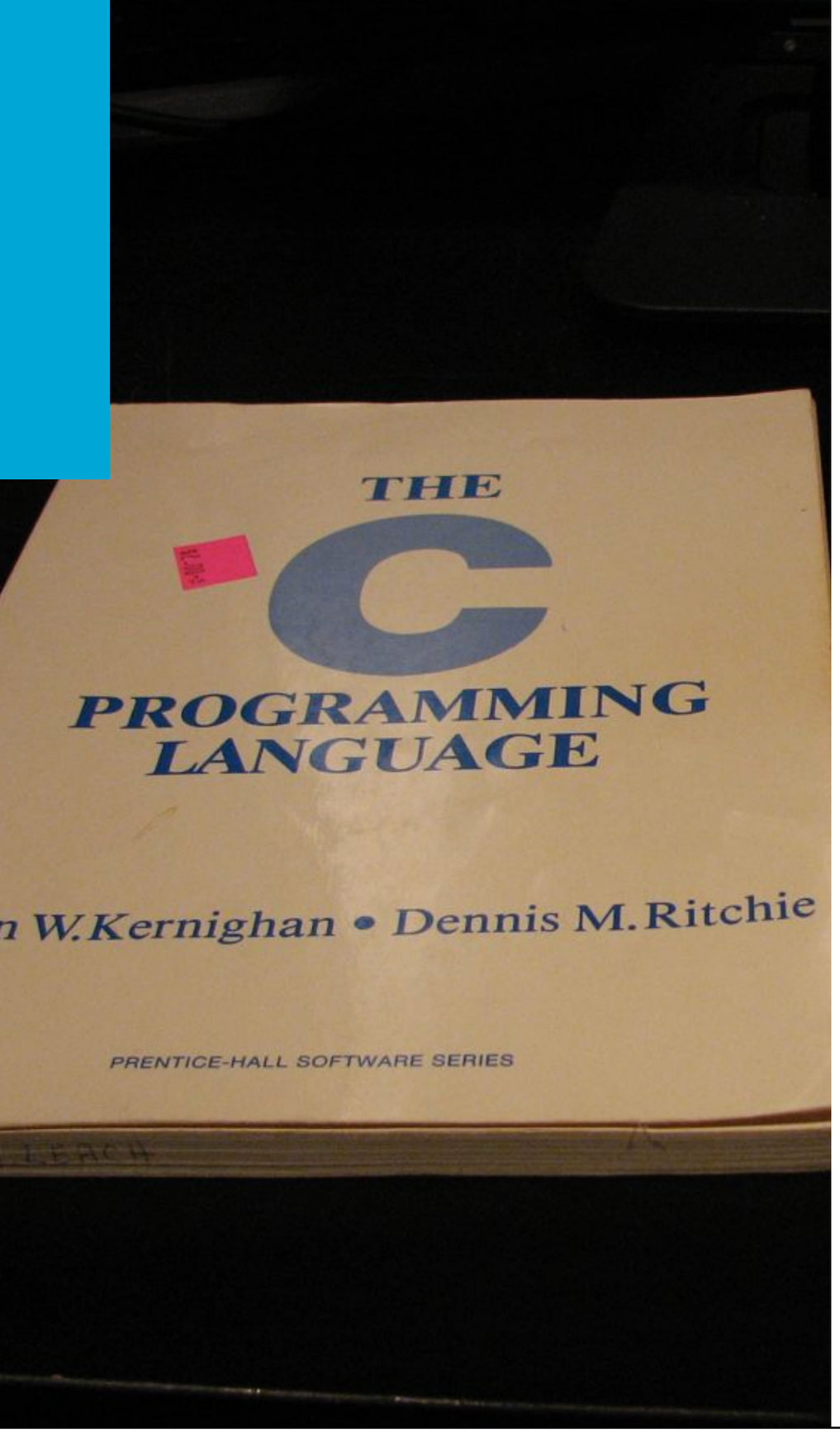
syntax

semantics

Tiger

the lecture language

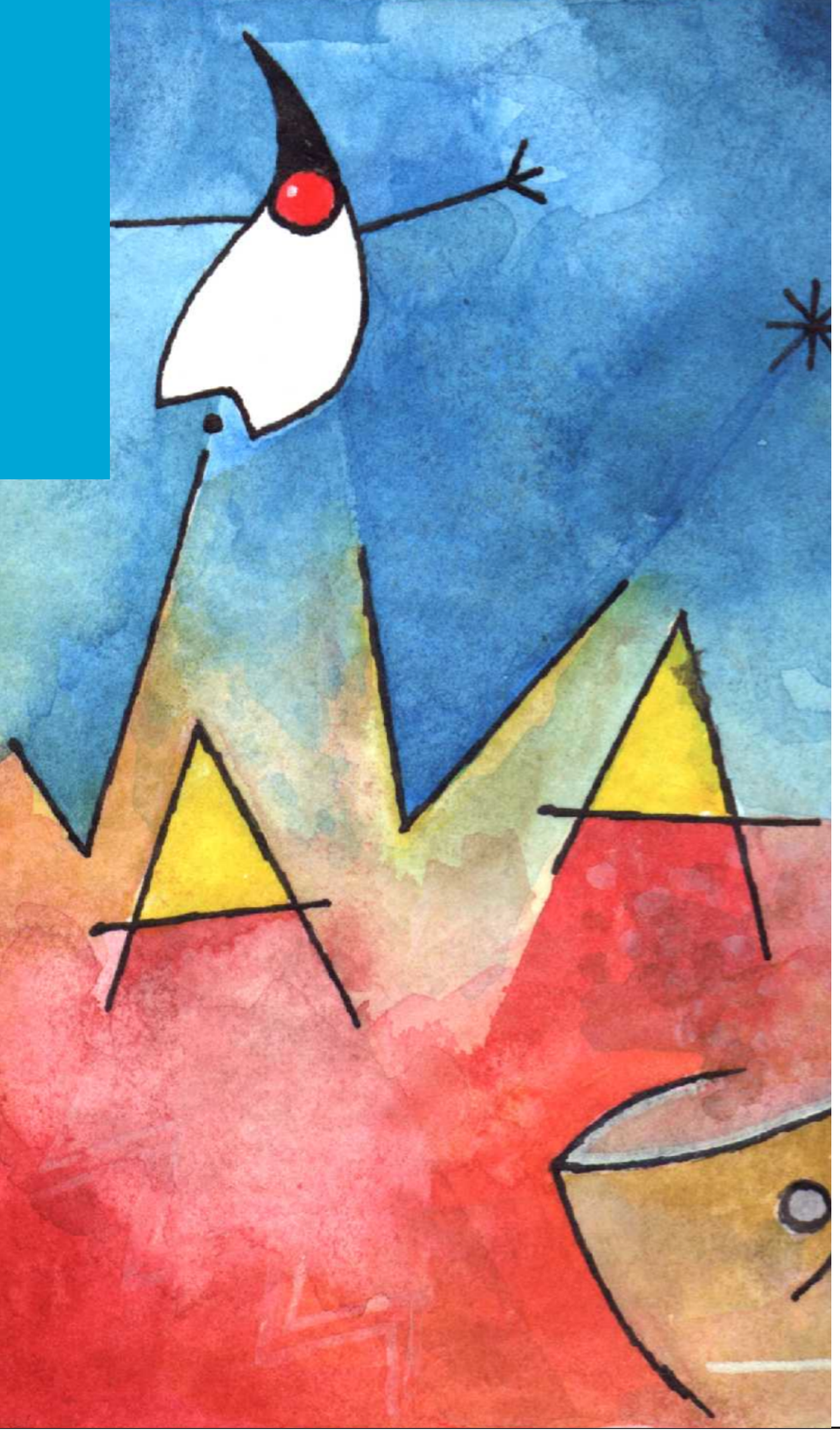
```
/* factorial function */

let

  var x := 0

  function fact(n : int) : int =
    if n < 1 then 1 else (n * fact(n - 1))

in

  for i := 1 to 3 do (
    x := x + fact(i);
    printint(x);
    print(" ")
  )

end
```



C

another lecture language

```c
#include <stio.h>

/* factorial function */

int fac(int num) {
  if (num < 1)
     return 1;
  else
    return num * fac(num - 1);
}

int main() {
  printf("%d! = %d\n", 10, fac(10));
  return 0;
}
```

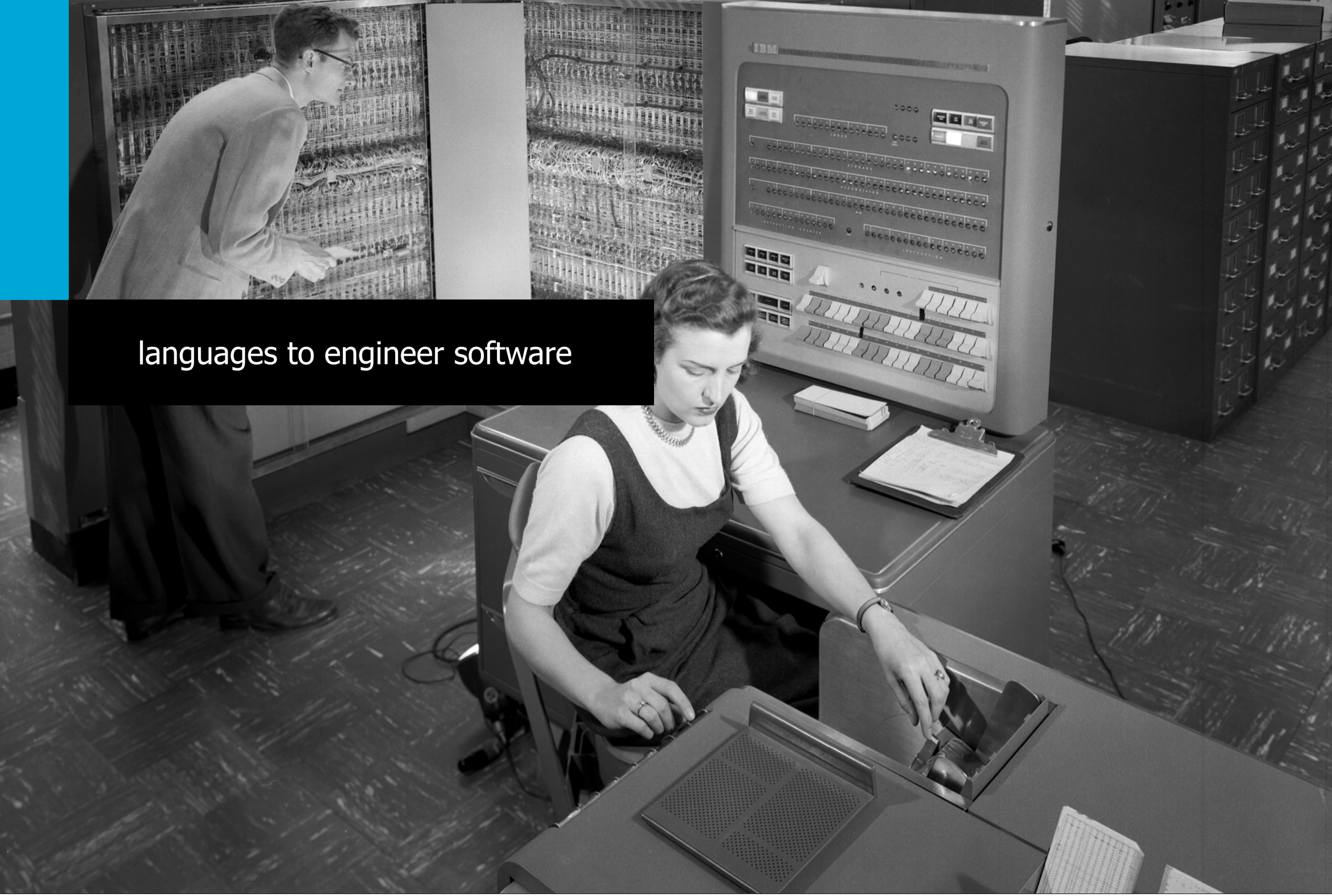


MiniJava

the lab language

```
class Main {

  public static void main(String[] args) {
    System.out.println(new Fac().fac(10));
  }
}

class Fac {

  public int fac(int num) {
    int num_aux;
    if (num < 1)
      num_aux = 1;
    else
      num_aux = num * this.fac(num - 1);
    return num_aux;
  }
}
```

language software



languages to engineer software

pieces of software themselves

to provide a means

for realising a stated process

on a variety of machines

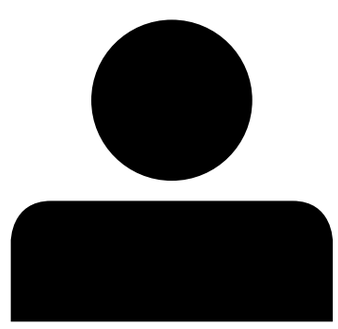
John W. Backus: The Syntax and Semantics of the Proposed International Algebraic Language of Zürich ACM-GAMM Conference.

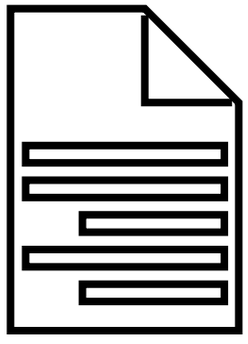
to provide a means

for **realising** a **stated process**

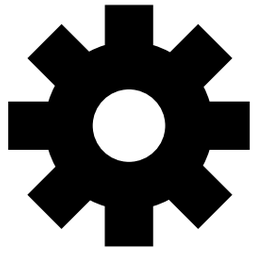
on a variety of **machines**

John W. Backus: The Syntax and Semantics of the Proposed International Algebraic Language of Zürich ACM-GAMM Conference.

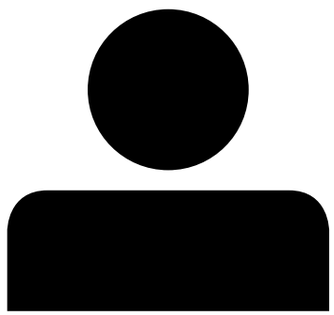


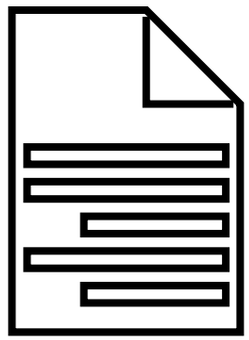


software language

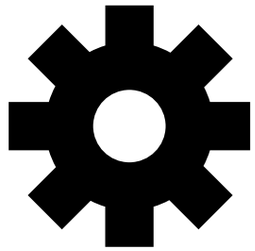


interpreter

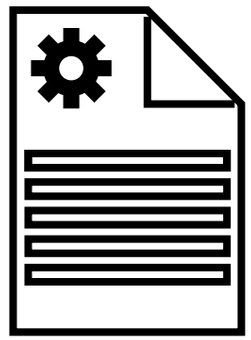




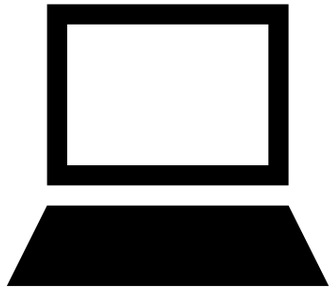
software language

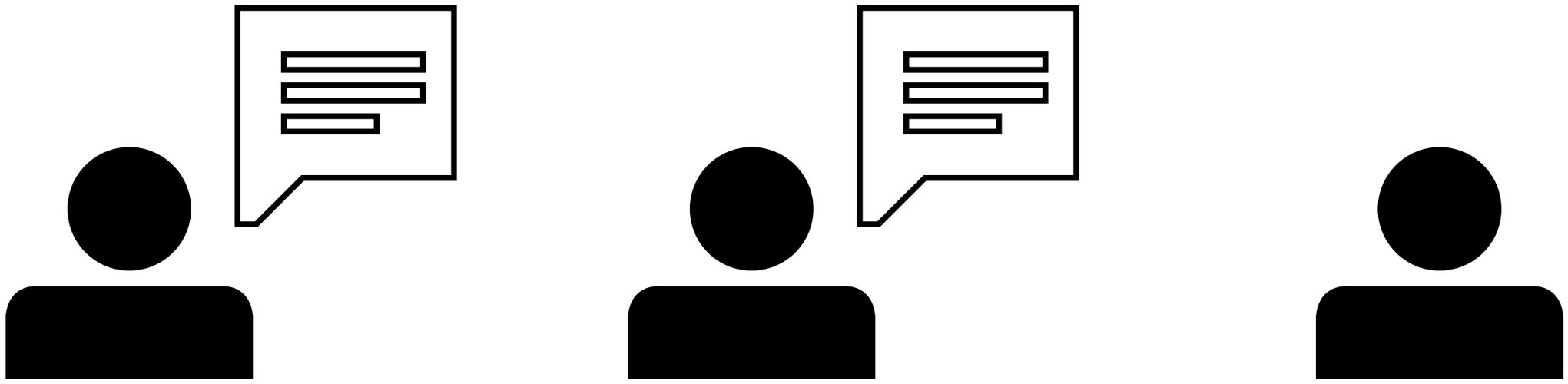


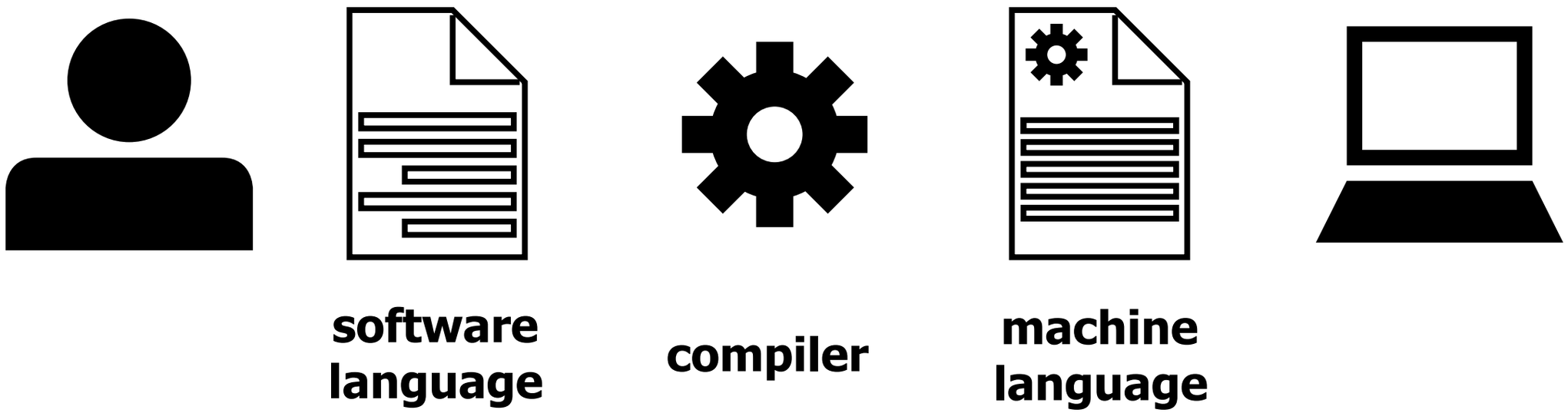
compiler



machine language




natural language
translator
natural language

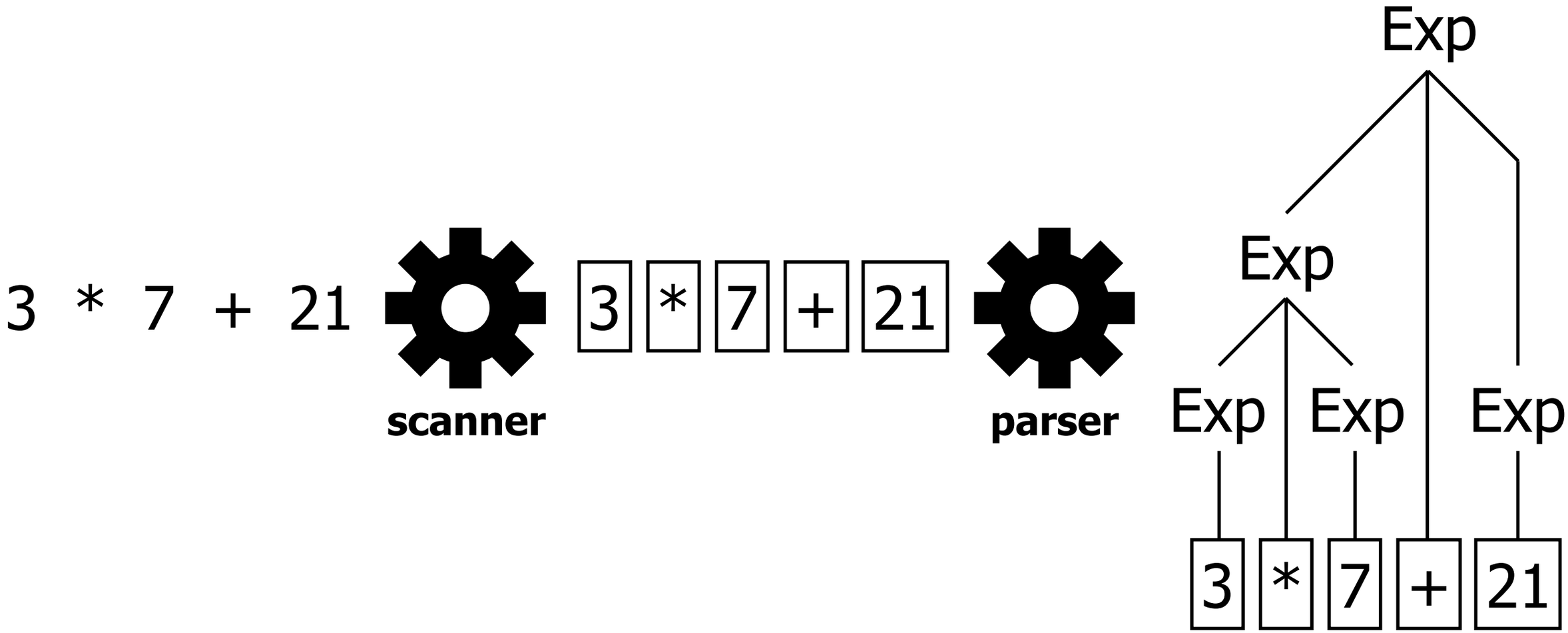

software language
compiler
machine language

language software

language processors

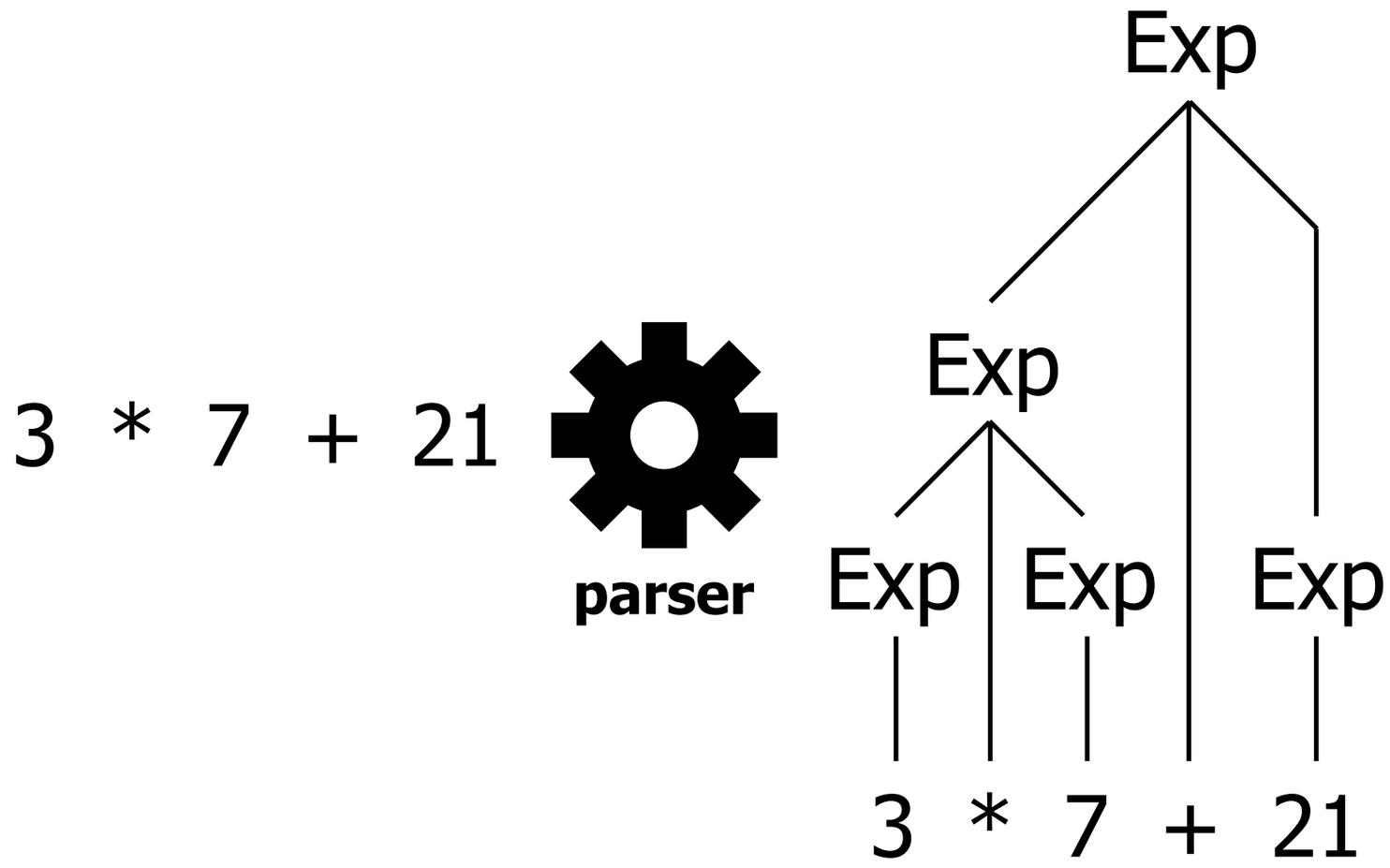
language processors

scanners & parsers



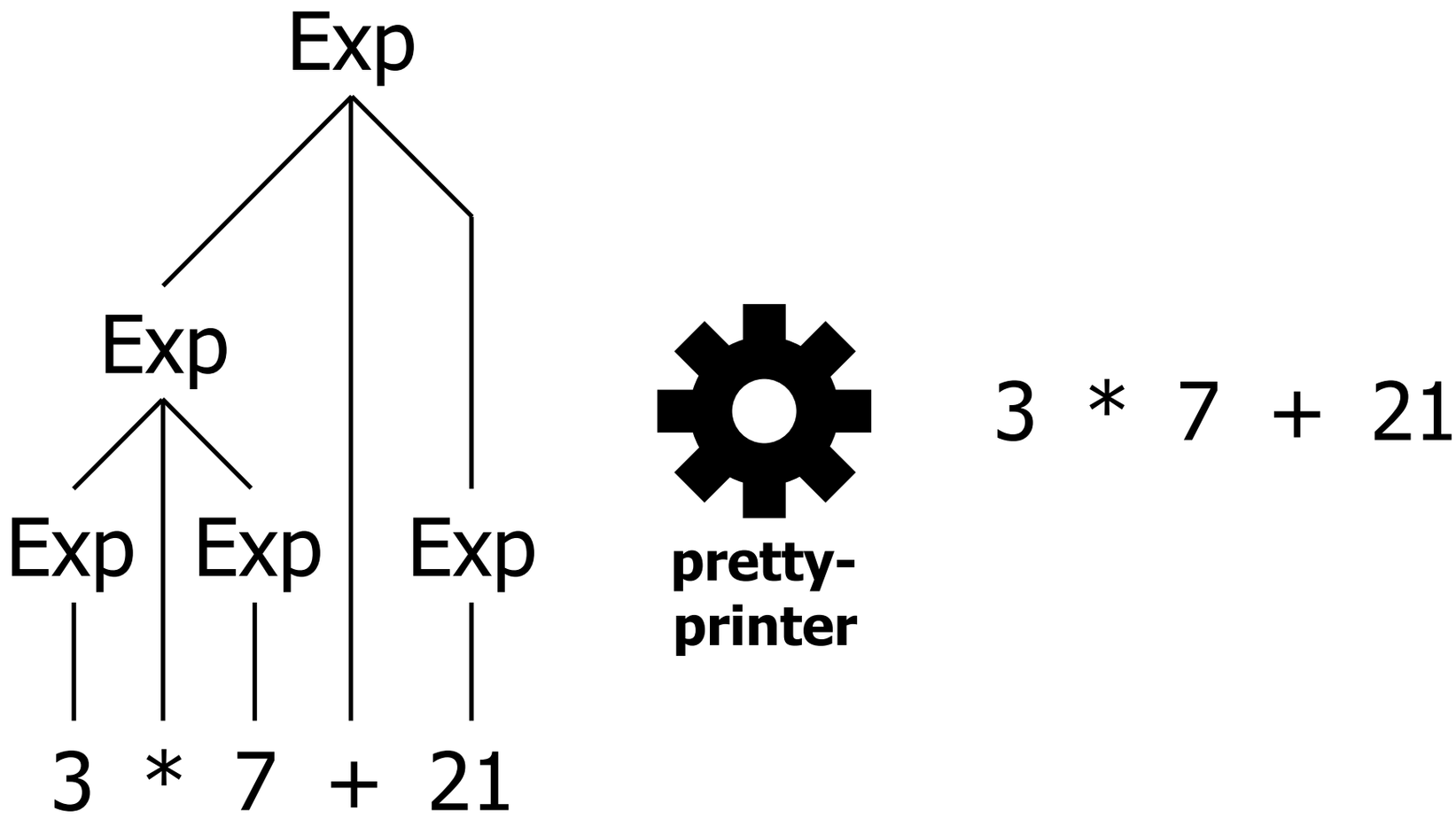
language processors

scannerless parsers



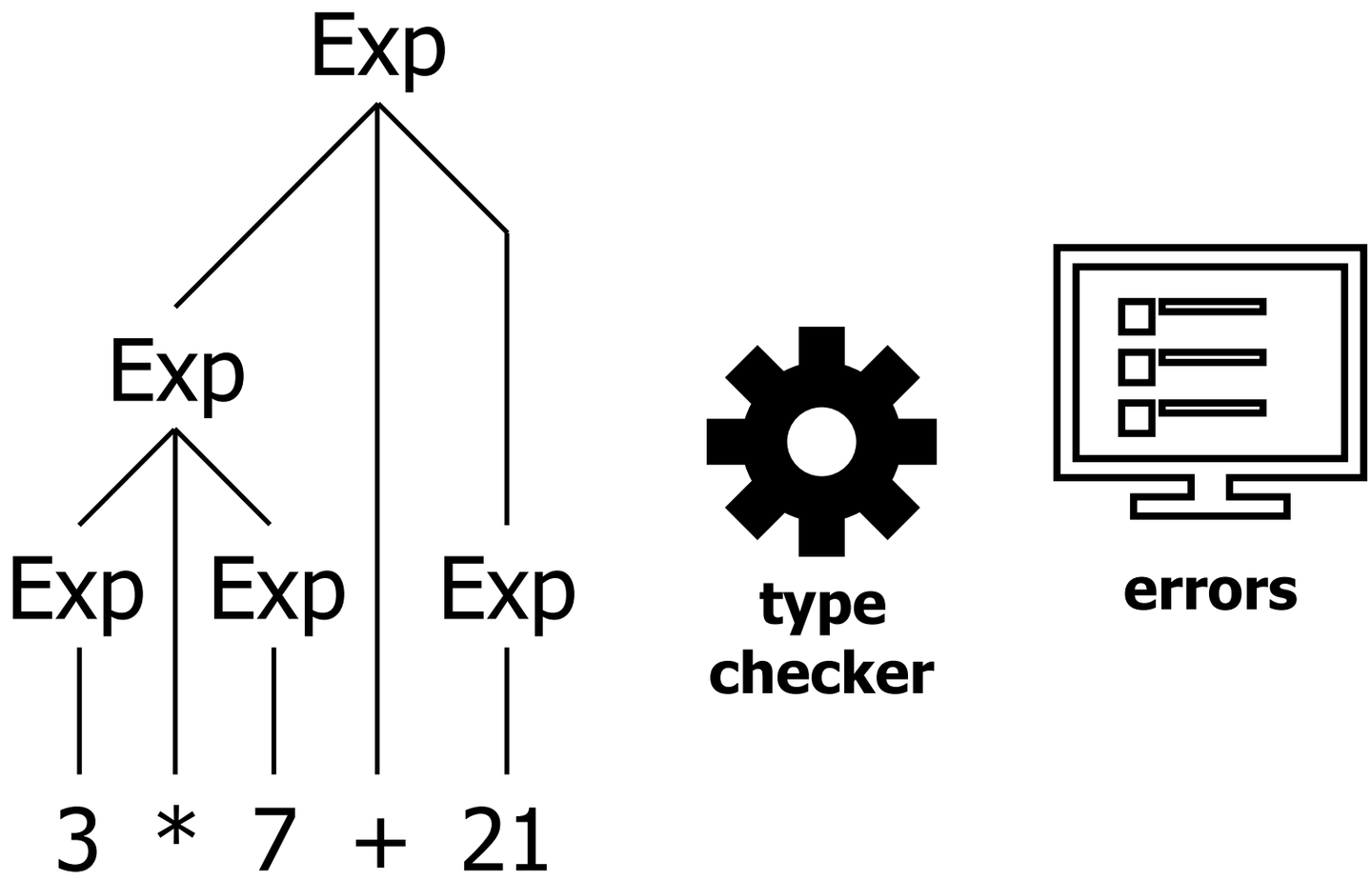
language processors

pretty-printers



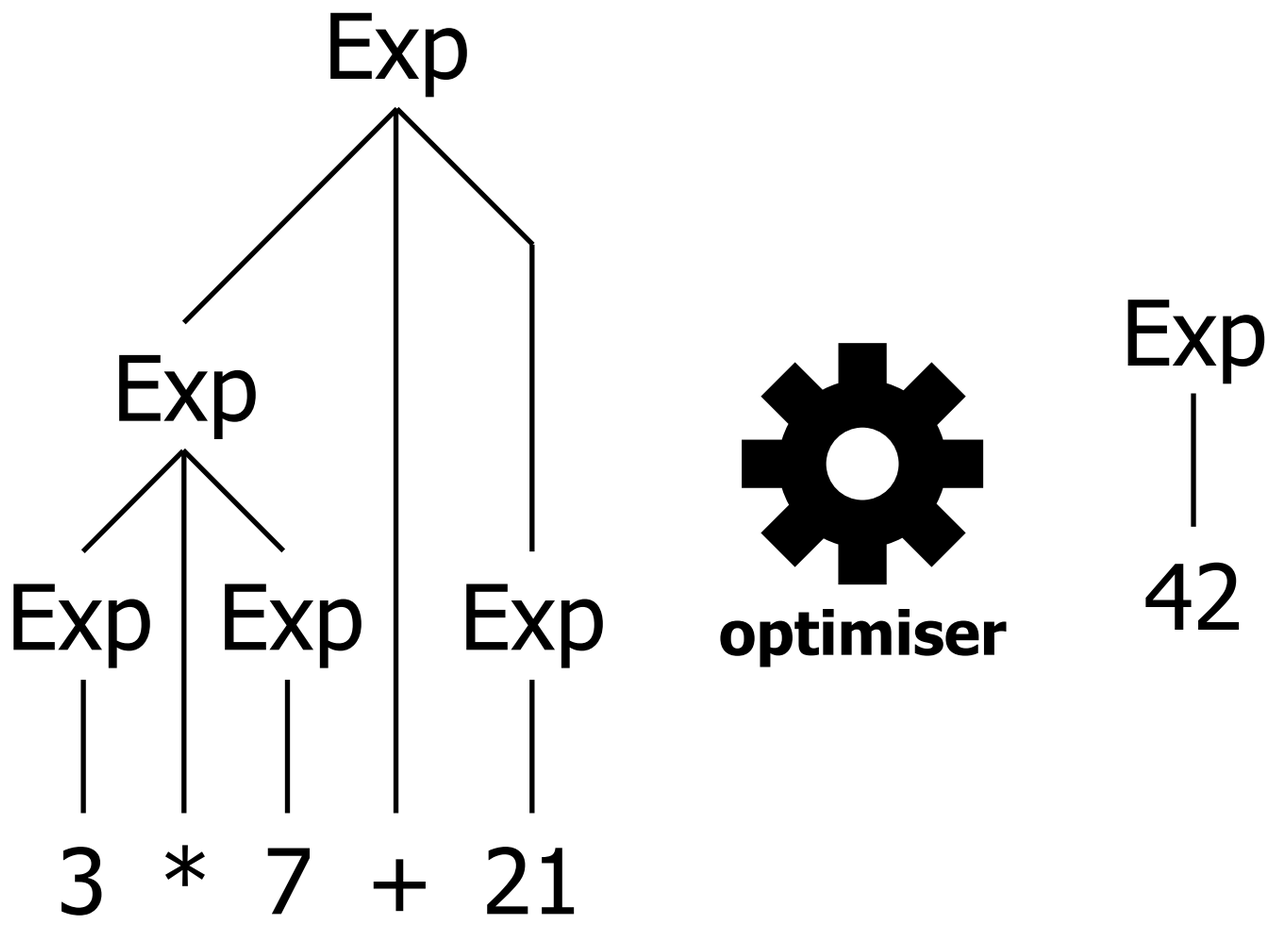
language processors

type checkers



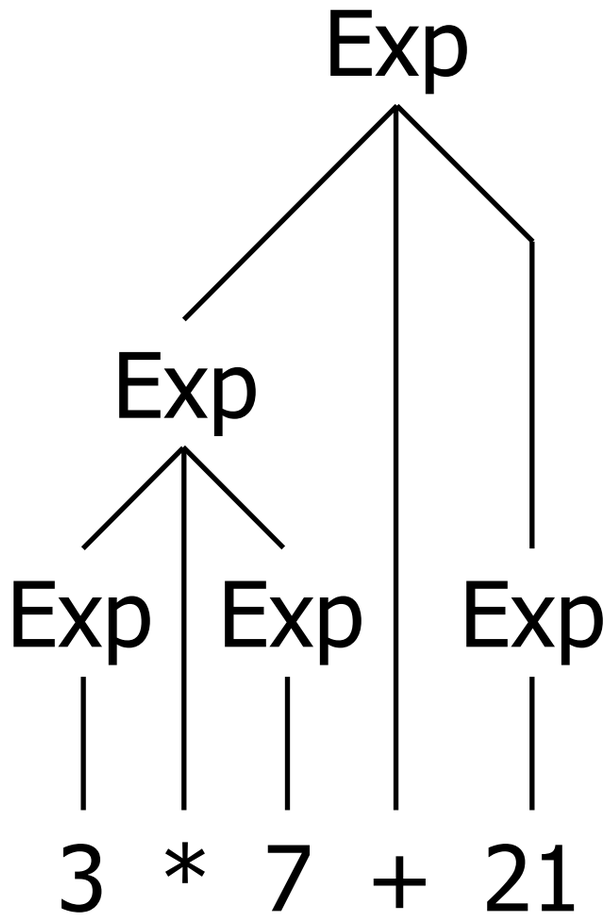
language processors

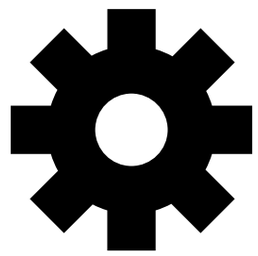
optimisers



language processors

generators






```
ldc   3
ldc   7
imul
ldc   21
iadd
```

language software

compilers

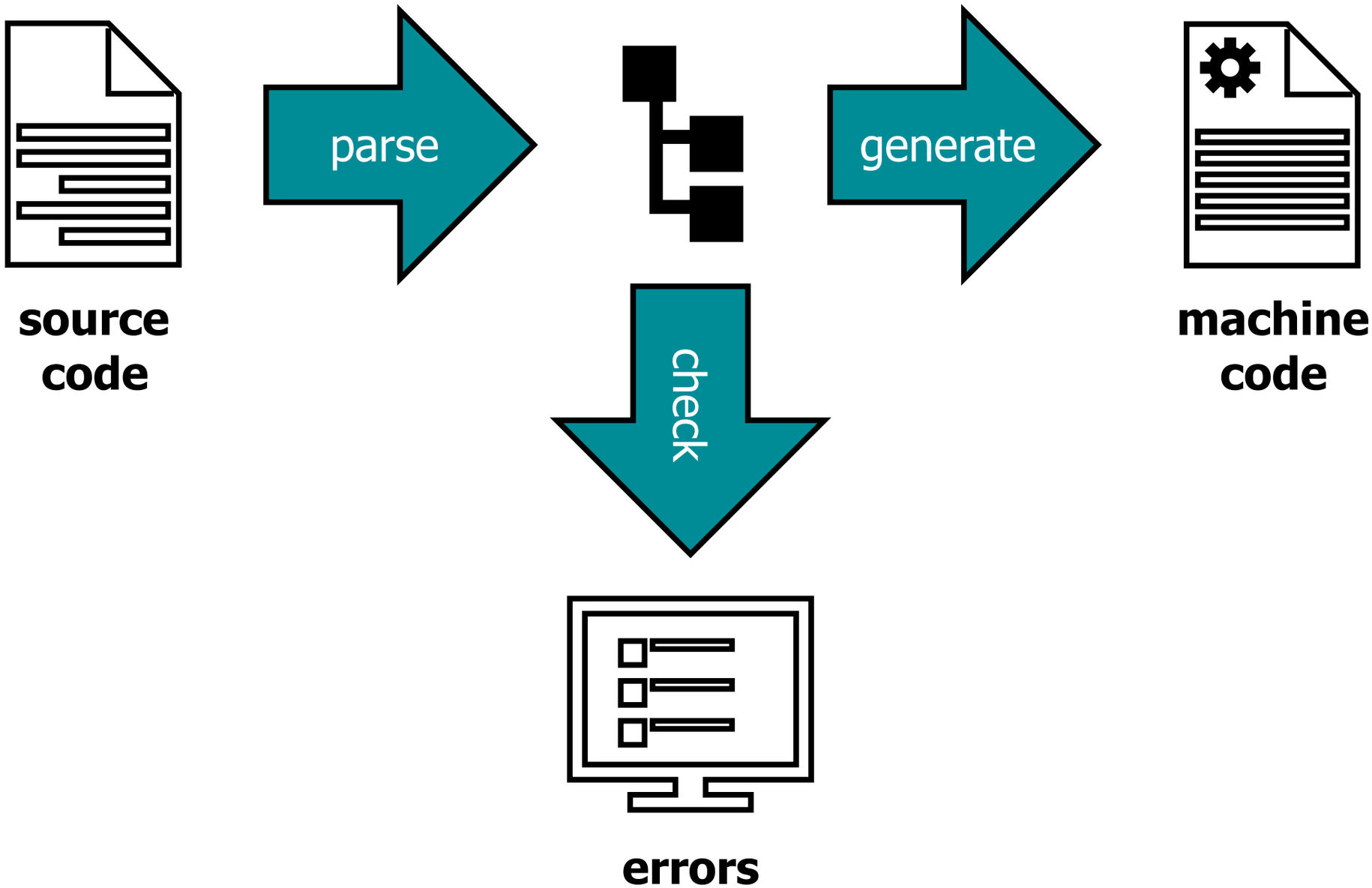
traditional compilers

example

```
> ls
    Course.java
> javac -verbose Course.java
    [parsing started Course.java]
    [parsing completed 8ms]
    [loading java/lang/Object.class(java/lang:Object.class)]
    [checking university.Course]
    [wrote Course.class]
    [total 411ms]
> ls
    Course.class      Course.java
```

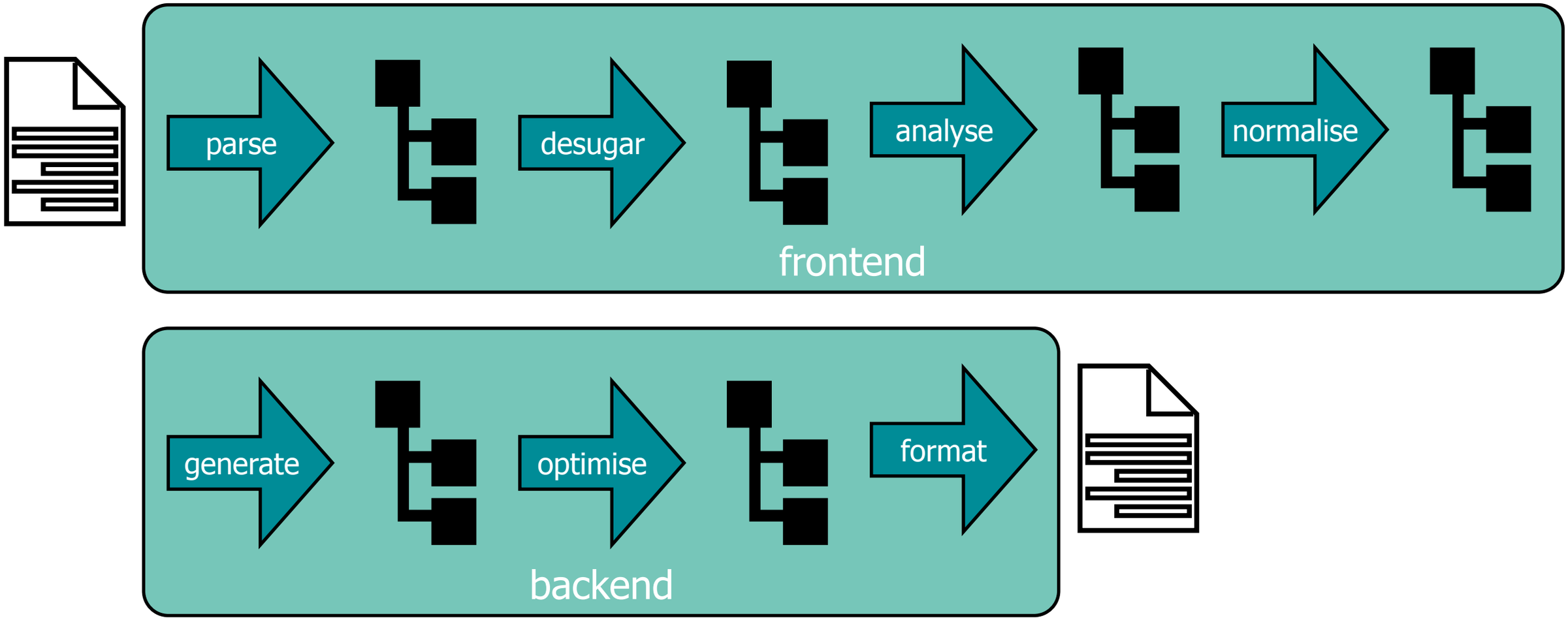
traditional compilers

architecture



traditional compilers

compilation by transformation



modern compilers in IDEs

features

```
1  class User {
2    string name;
3  }
4  class Blog {
5    string post(User user, string message) {
6      posterName = "name";
7      string posterName;
8      posterName = user.nam;
9      string posterName = user.name;
10     return posterName;
11   }
12 }
```

modern compilers in IDEs

features

```
1  class User {
2    string name;
3  }
4  class Blog {
5    string post(User user, string message) {
6      string posterName;
7      posterName = user.name;
8      return posterName;
9    }
10 }
```

```
1  class User {
2    string name;
3  }
4  class Blog {
5    string post(User user, string message) {
6      string posterName;
7      posterName = user.name;
8      return posterName;
9    }
10 }
```

modern compilers in IDEs

features

```
1  class Blog {
2    string post(User user, string message) {
3      string posterName;
4      posterName = user.;
5      return posterName;
6    }
7  }
8  class User {
9    string name;
10   string username;
11   string homepage;
12 }
```

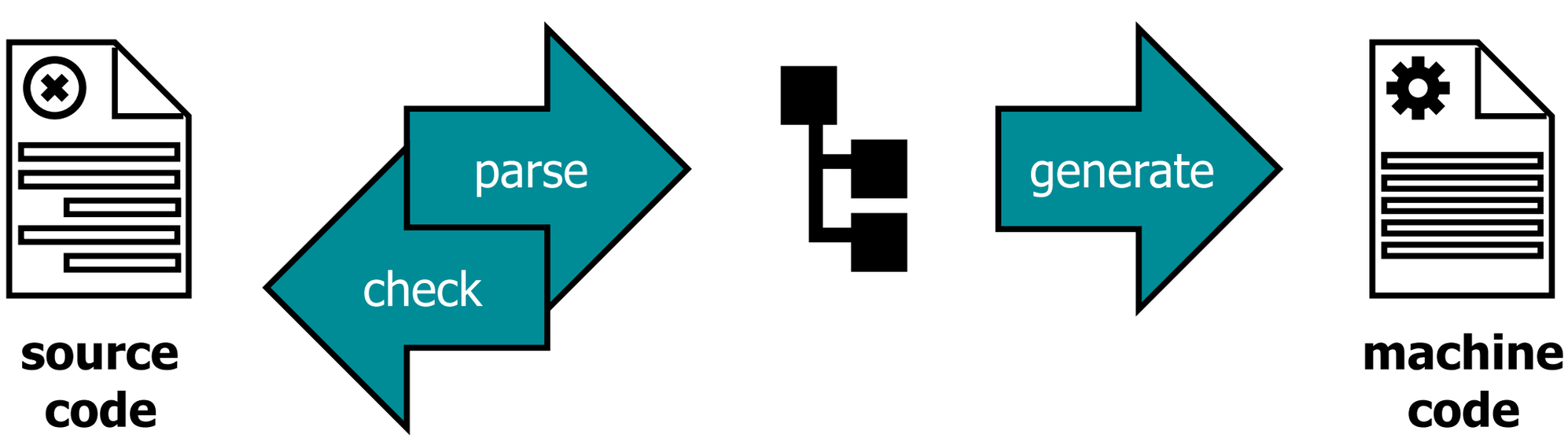
homepage
name
username

```
1  class Blog {
2    string post(User user, string message) {
3      string posterName;
4      posterName = user.name;
5      return ;
6    }
7  }
8  class User
9    string na
10   string us
11   string ho
12 }
```

message
posterName
user

modern compilers in IDEs

architecture



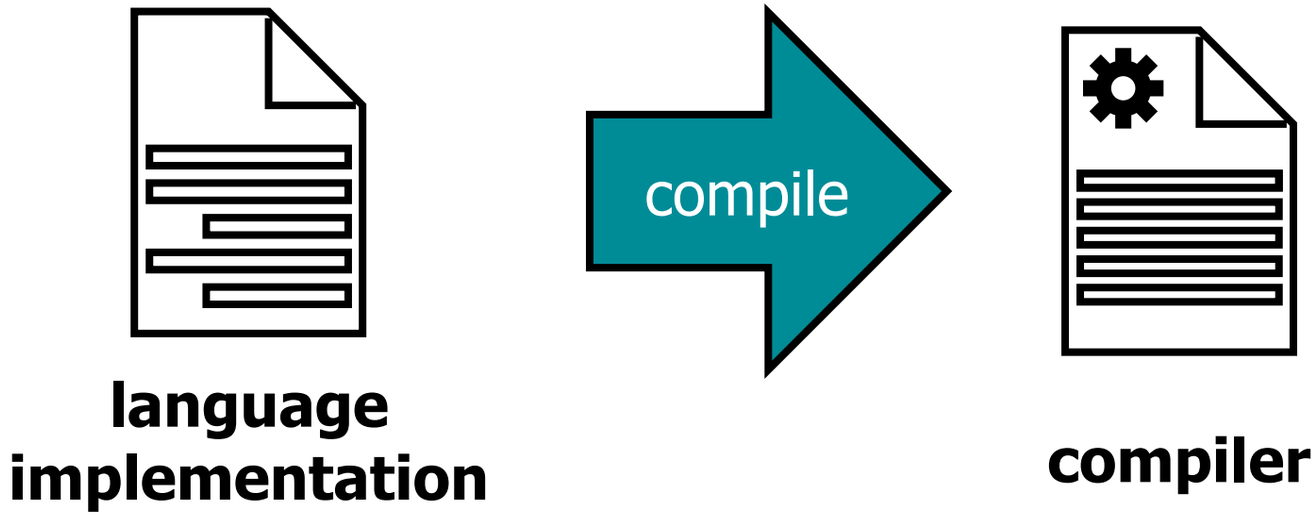
language software

compiler construction

meta language facility

traditional compiler compilers

manual implementation

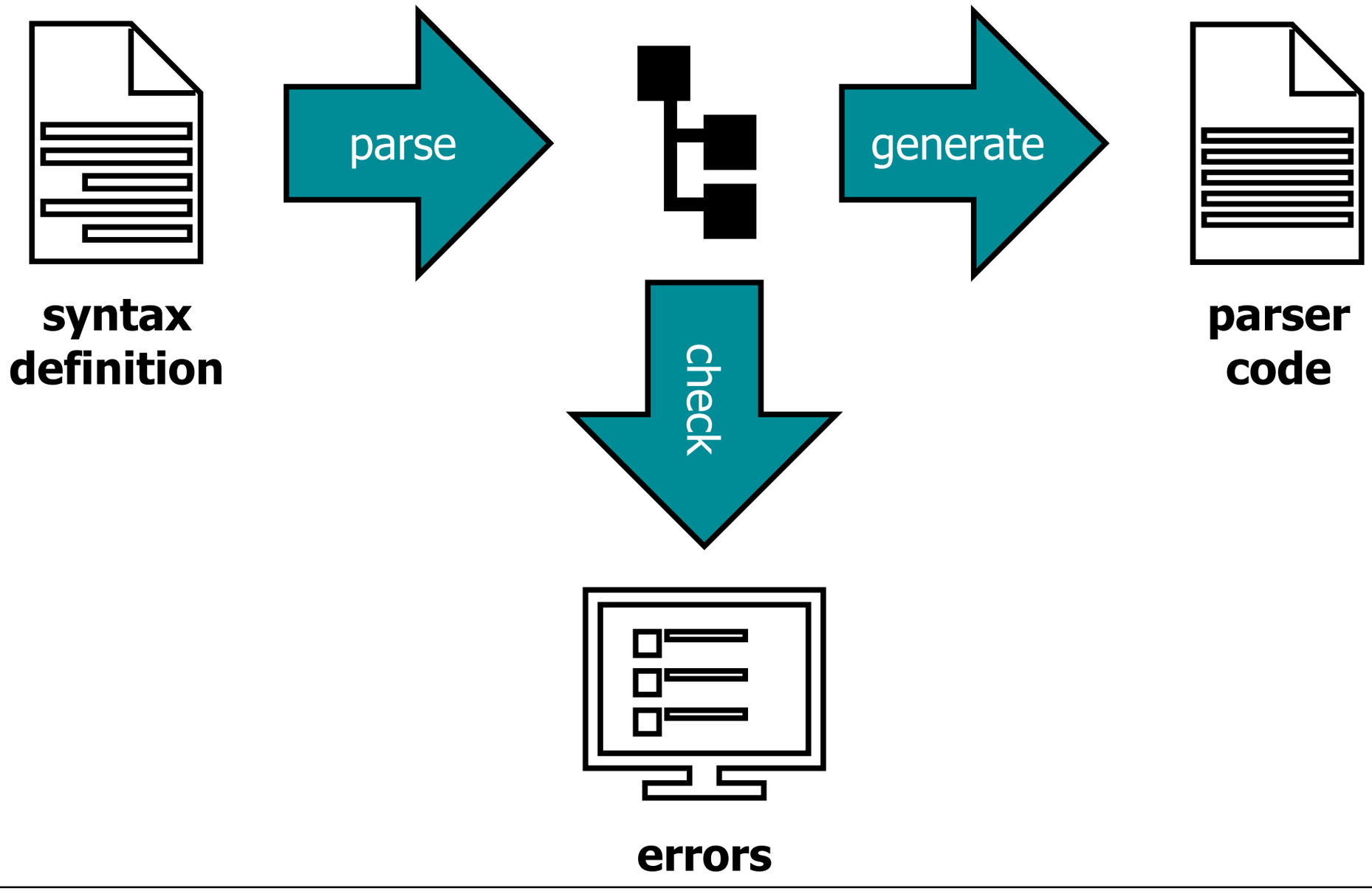


traditional compiler compilers

compilation + compilation

traditional compiler compilers

example

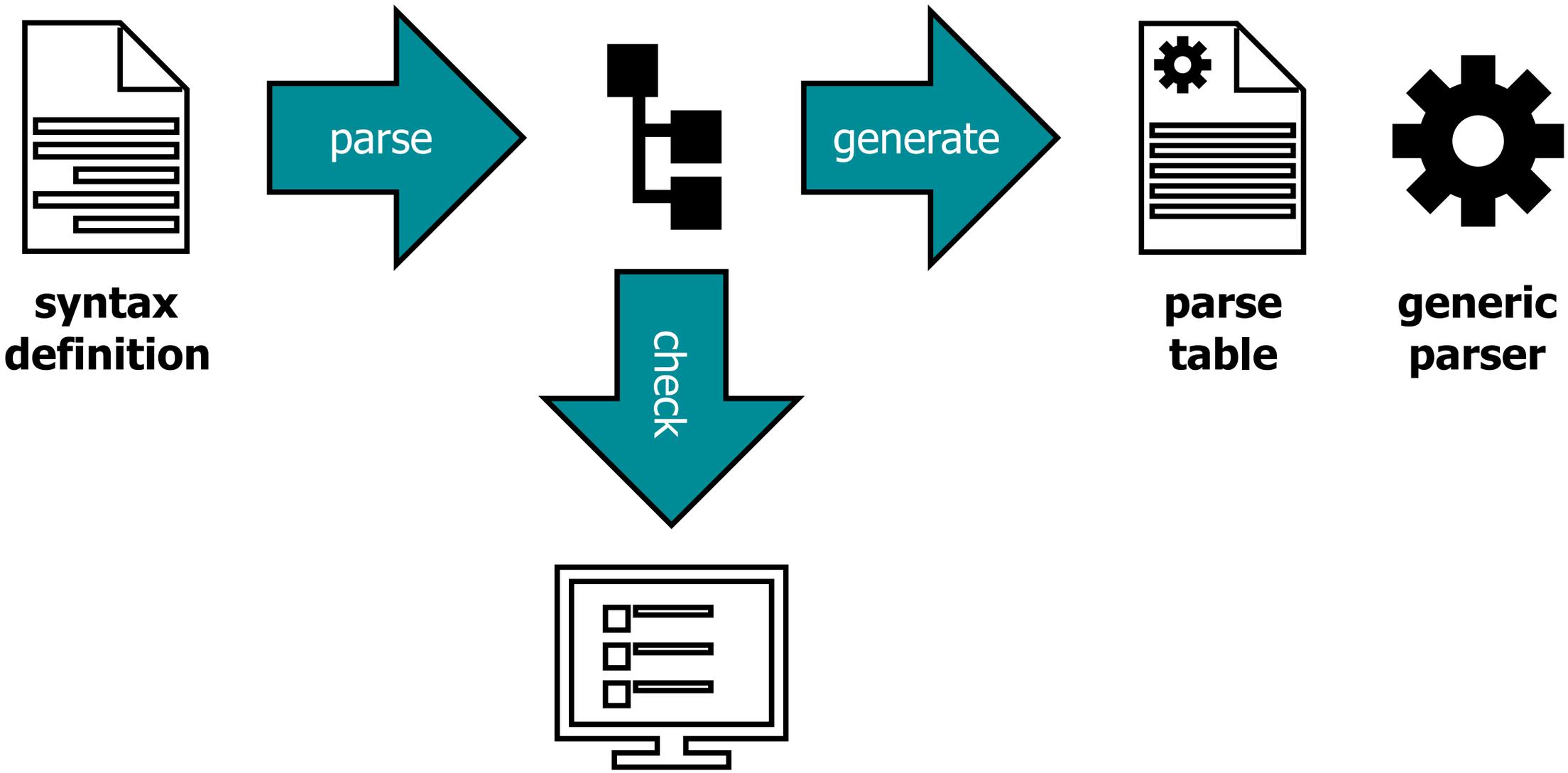


traditional compiler compilers

compilation + interpretation

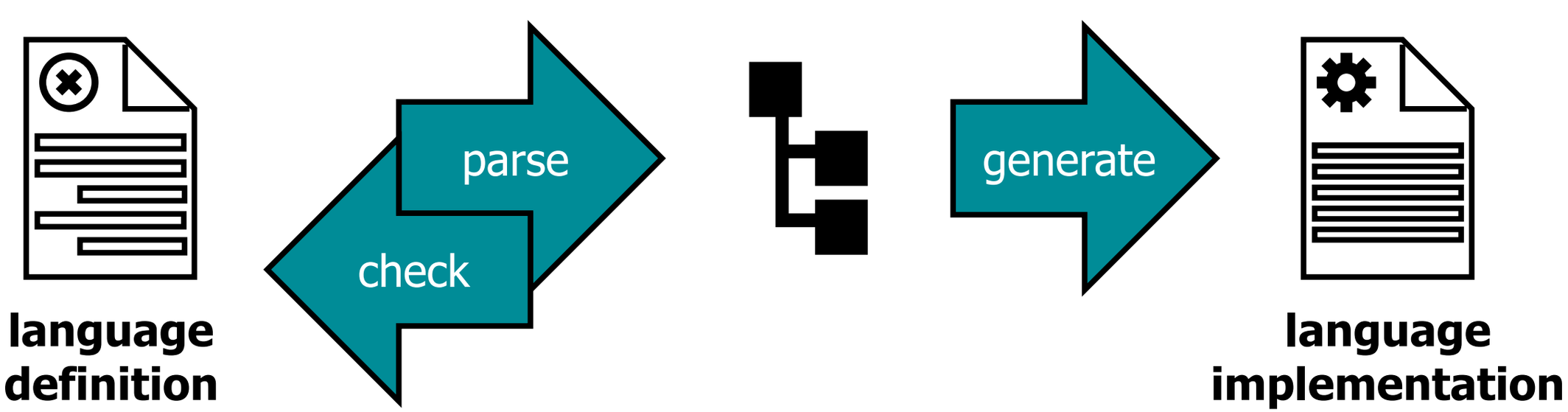
traditional compiler compilers

example

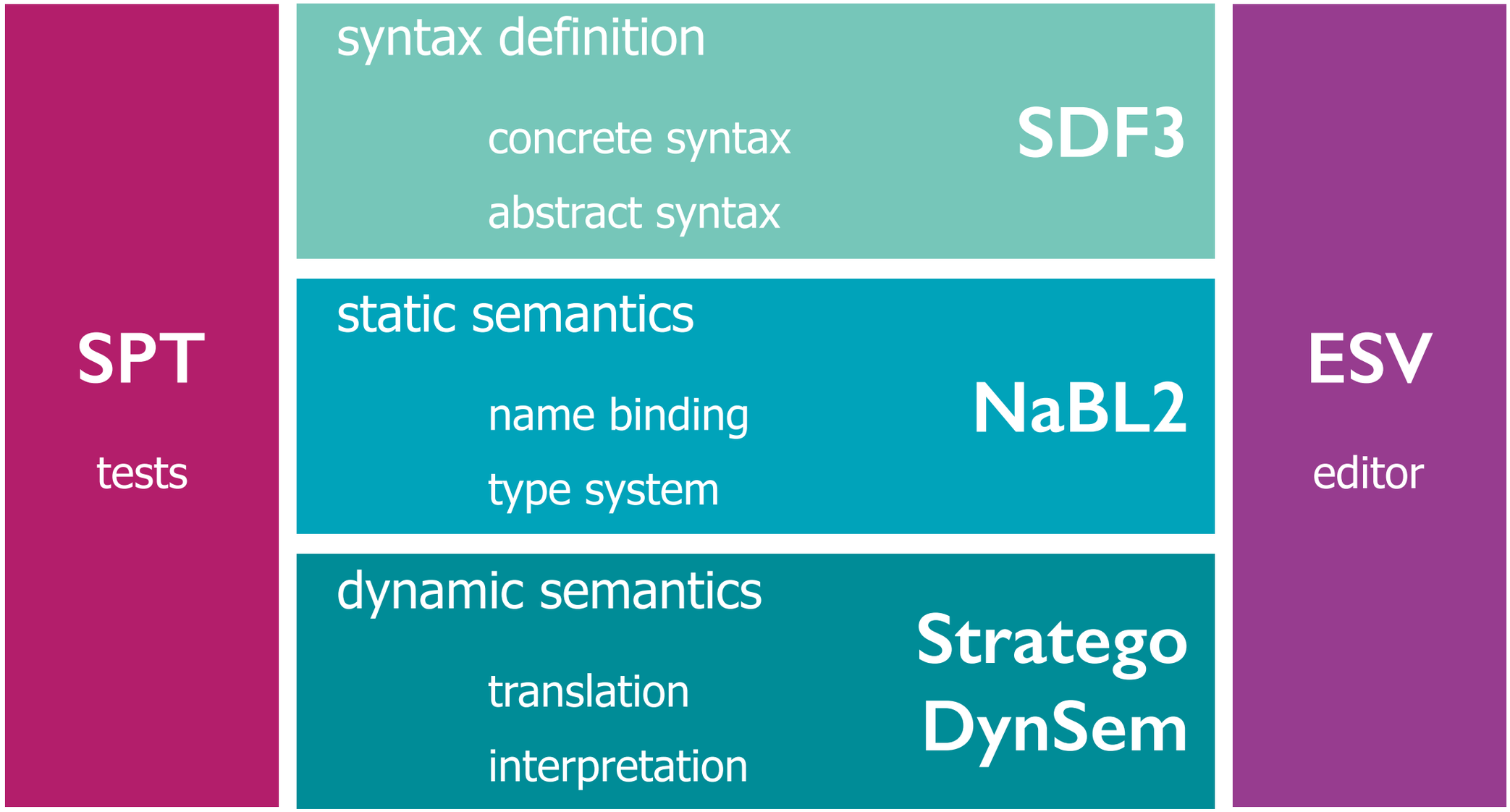


language workbenches

architecture



Spoofax language workbench


SPT
tests
syntax definition
concrete syntax
abstract syntax
SDF3
static semantics
name binding
type system
NaBL2
dynamic semantics
translation
interpretation
Stratego
DynSem
ESV
editor

attribution

slide	title	author	license
1, 35	Programming language textbooks	K.lee	public domain
2	The Bigger Picture	F. Delventhal	CC BY 2.0
5	Rose 2 des Kleinen Prinzen	Antoine Saint-Exupéry	public domain
11	Dog and owner on Ballykinler Beach	Jonny Green	CC BY 2.0
12	Fashionable melange of English words (1)	trialsanderrors	public domain
13	Gift	asenat29	CC BY 2.0
13	Toxic	John Morgan	CC BY 2.0
14	Latin Grammar	Anthony Nelzin	
15	Ginkgo Biloba	Johann Wolfgang von Goethe	public domain
16	Sub-deb slang	genibee	CC BY-NC 2.0
17	Wednesday	Michael Fawcett	CC BY-NC-SA 2.0
19, 25, 67	Thesaurus	Enoch Lau	CC BY-SA 3.0
22	The captured Swiftsure, Seven Oaks, Loyal George and Convertine brought through Goeree Gat	Willem van de Velde the Younger	public domain

attribution

slide	title	author	license
23	Tower of Babel	Pieter Bruegel the Elder	public domain
26	The Ashtadhyayi	translated by Srisa Chandra Vasu	public domain
27	Buchdruckerduden		public domain
28	Boeing 737-300 -400 -500 Manitenance Manual Alaska Airlines	Bill Abbott	CC BY-SA 2.0
30	Esperanto		public domain
31	Quenya	Juanma Pérez Rabasco	CC BY 2.0
32	Klingon Dictionary	Josh Bancroft	CC BY-NC 2.0
33	Overweight Na'vi	HARRY NGUYEN	CC BY 2.0
36, 45, 46	IBM Electronic Data Processing Machine	NASA	public domain
41	Tiger	Bernard Landgraf	CC BY-SA 3.0
42	The C Programming Language	Bill Bradford	CC BY 2.0
43	Italian Java book cover		
49-73	PICOL icons	Melih Bilgil	CC BY 3.0