

CHOOSE FORUM 2017

Daring to do projects other don't dare to dream

Darius Blasband
darius@raincode.com
Zurich
November 2017

1 slide presentation of the company

- Based in Brussels, Belgium
 - Small operation in Bangalore, India
 - Sales office in Jacksonville, Florida
- Founded in 1998
 - >50 people
 - Privately owned

We're hiring!

“Compiler mercenaries”
“Accenture for compilers”
“All things programming languages”



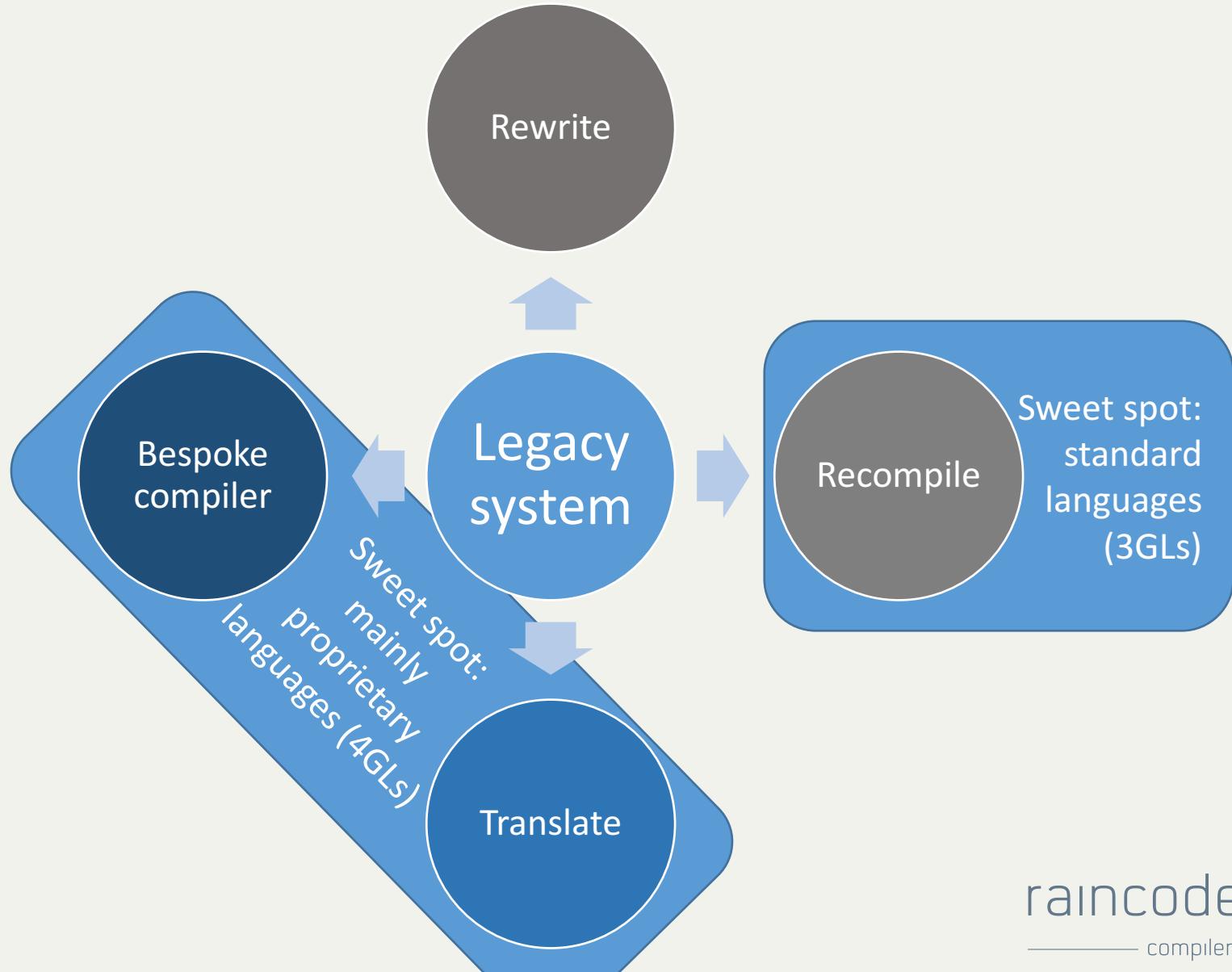
A. Time



Quiz:

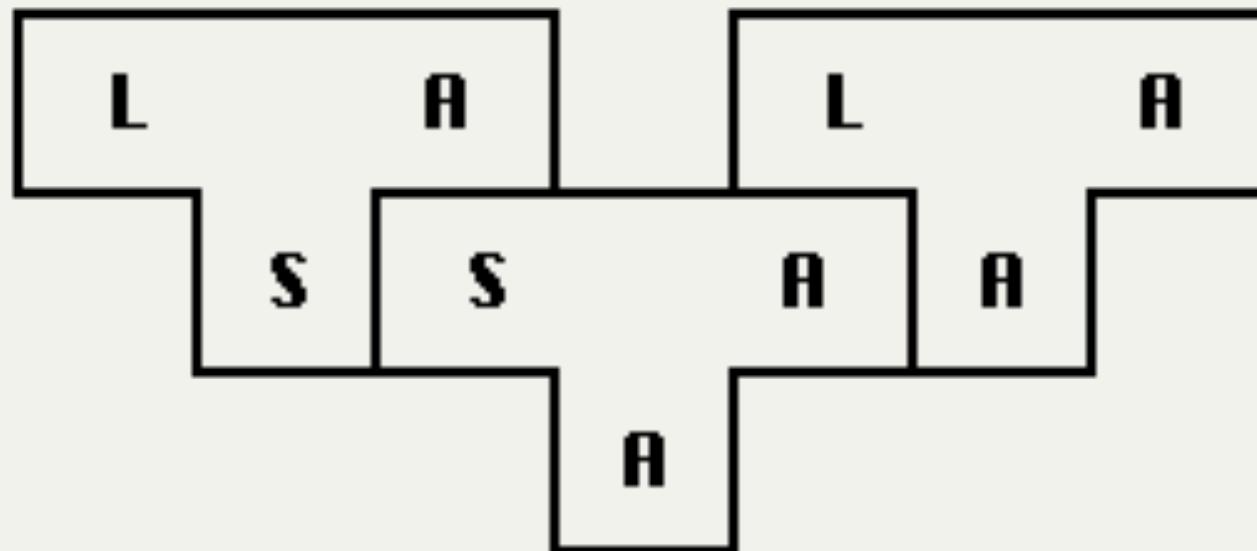
What language would you use
for a system that must live for at
least 30 years ?

How to best deal with a legacy system?



From a distance, translation vs. compilation...

- It's all the same!
- It's a translation from a language to another after all



In practice though:

Translation

- Maintainability
 - Structure
 - Possibly incomplete
 - Possibly inaccurate
 - Limit entropy
 - Sold as a service
 - Limited term commitment and relationship
 - Two guys in a garage
- 

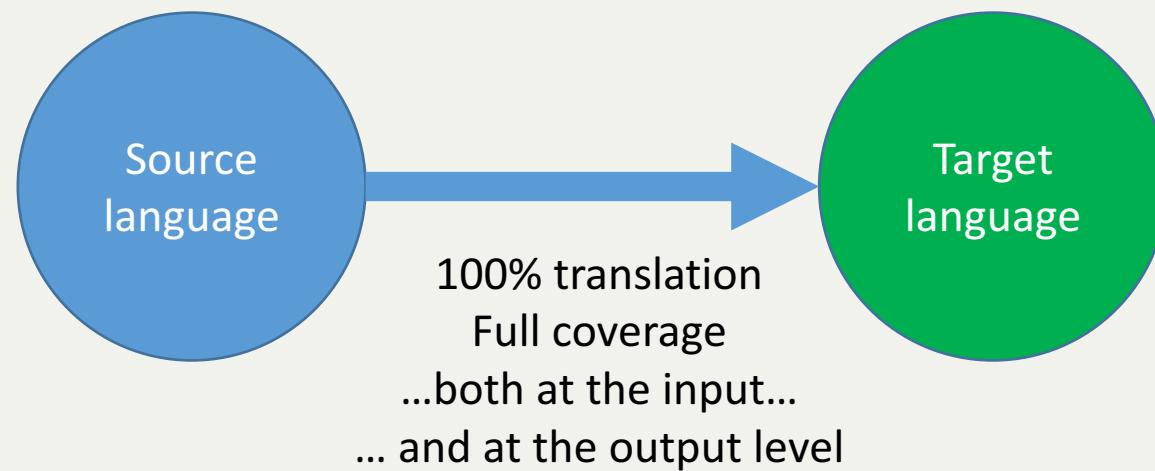
Compilation

- Executability
 - Performance
 - Completeness
 - Accuracy
 - Entropy is irrelevant
 - Sold as a product
 - Long term commitment and relationship
 - Real company
- 

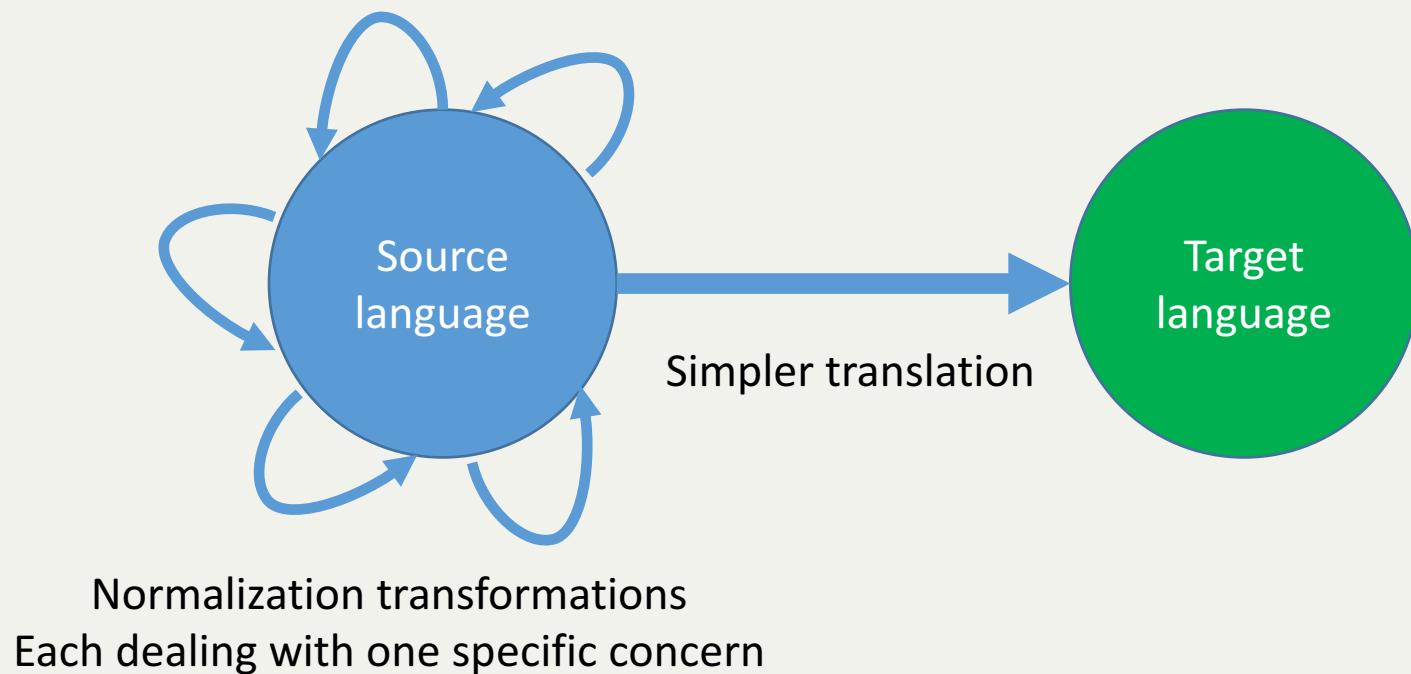
B. Transformation



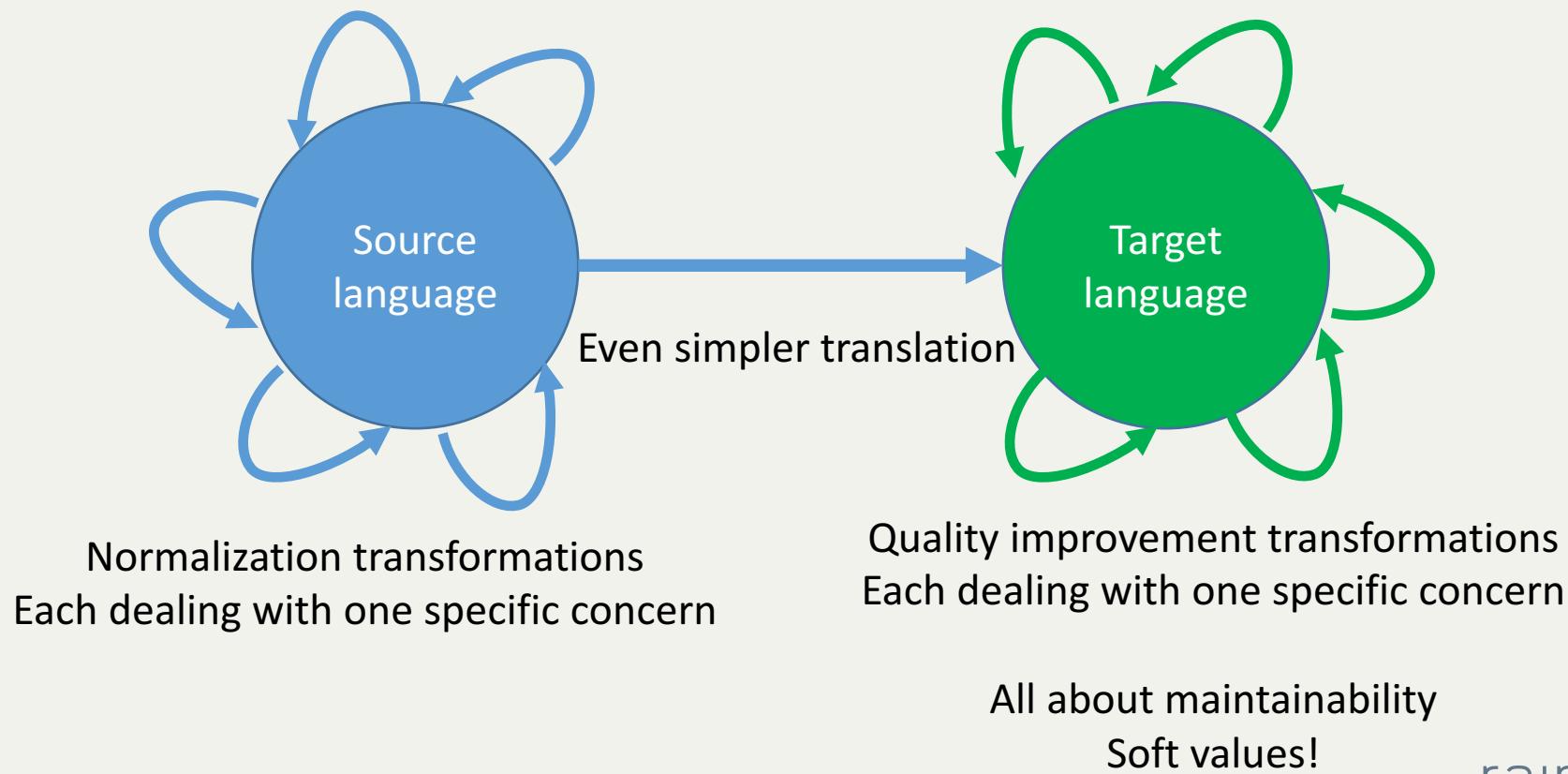
The process, as understood naively



The process, in a more realistic setting



Symmetry



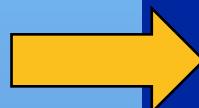


Example in transformation:

```
F05DC.
    MOVE      1                      TO ICATR.      (1)
    GO TO    F05DC-B.
F05DC-A.
    ADD      1                      TO ICATR.      (2)
F05DC-B.
    IF       10                     < ICATR THEN (3)
        GO TO    F05DC-FN
    END-IF.
    IF       CATX(ICATR) = '0' THEN
        NEXT SENTENCE
    ELSE
        GOTO F05DC-C
    END-IF.
    MOVE 'X' TO CATM(ICATR).      (5)
F05DC-C.
    IF       CATX(ICATR) NOT = '0' THEN
        NEXT SENTENCE
    ELSE
        GOTO F05DC-D
    END-IF.
    MOVE 'Y' TO CATM(ICATR).      (7)
F05DC-D.
    GO TO    F05DC-A.              (8)
F05DC-FN.
    EXIT.
```



Resulting code:



```
PERFORM WITH TEST BEFORE          (8)
      VARYING ICATR FROM 1          (1)
                           BY 1          (2)
      UNTIL 10 < ICATR            (3)
      IF CATX(ICATR) = '0' THEN    (4)
          MOVE 'X' TO CATM(ICATR)  (5)
      ELSE                         (6)
          MOVE 'Y' TO CATM(ICATR)  (7)
      END-IF
END-PERFORM.
```

- Smaller
- Intention is clear
- No more GOTOs, no more labels
- Better performance
- 100% Safe

```

EVALUATE TRUE
WHEN GC06-QPSTD NUMERIC AND GC06-QPSTD > ZEROES
    MOVE KB00-QSHOWP      TO 7-WORK-ATRAR
WHEN GC06-PPOTD NUMERIC AND GC06-PPOTD > ZEROES
    IF KB00-ICKPY = 'Y' THEN
        MOVE KB00-PACT1      TO 7-WORK-ATRAR
    ELSE
        MOVE 7-FULL-PERCENT  TO 7-WORK-ATRAR
    END-IF
END-IF

EVALUATE TRUE
WHEN GC06-ACOTD NUMERIC AND GC06-ACOTD > ZEROES
    MOVE KB70-AEDRQ       TO 7-WORK-ATRAR
    AND GO
END-EVALUATE
    AND KB00-ICKPY = 'Y'
MOVE KB00-PACT1 TO 7-WORK-ATRAR
WHEN GC06-PPOTD NUMERIC
    AND GC06-PPOTD > ZEROES
    AND KBX0-ICKPY NOT = 'Y'
MOVE 7-FULL-PERCENT TO 7-WORK-ATRAR
WHEN OTHER
    CONTINUE
END-EVALUATE

IF GC06-QPSTD NUMERIC
    AND GC06-QPSTD > ZEROES
    MOVE KB00-QSHOWP      TO 7-WORK-ATRAR
END-IF

> ZEROES THEN
    RAR
    > ZEROES THEN
    RAR
    > ZEROES THEN
    RAR
    > ZEROES THEN
    RAR
    END-IF
    END-IF
    END-IF
    ELSE
        IF GC06-ACOTD NUMERIC AND GC06-ACOTD > ZEROES THEN
            MOVE KB70-AEDRQ       TO 7-WORK-ATRAR
        END-IF
    END-IF
    END-IF

```

C. Compilation



When is a bespoke compiler for a legacy language a good idea?

- When a language does not translate gracefully to anything useful
 - Esoteric control flow
 - Esoteric ([COBOL](#)-inspired) data types
- When existing implementations are not adequate
 - Facebook's HPHPc and HHVM
- When the level of abstraction offers significant and unique added value
- When people just can't/won't consider change
- When the user base is spread out

People are much harder to deal with than technology



Equivalence more important than correctness

- a is a fixed bin (d,s) :

$$a = \frac{n}{2^s}$$

- If I want to convert a to a fixed dec (e,t) , I must find m such that:

$$a = \frac{m}{10^t}$$

$$a = \frac{n}{2^s} \cdot \frac{5^s}{5^s} = \frac{n \cdot 5^s}{10^s} = \frac{n \cdot 5^s}{10^{s-t}}$$

which yields:

$$m = \frac{n \cdot 5^s}{10^{s-t}}$$

Where the division by a power of 10 is trivial
(merely removing extra digits on the right)



The problem is:

- IBM indicates that its PL/1 compiler approximates

$$n \cdot 5^s$$

- by

$$n \cdot (5^s + 1)$$

- For some unspecified reason

It is clear that this bias is only introduced for large values of s.

On smaller values, there is no point in using approximations anyway

After a few experiments, it seems that the bias is introduced for $s \geq 14$

Which makes sense as 14 is the smallest value of s such that:

$$|\log_2 5^s| > 31$$

- We reproduce this (odd) behaviour scrupulously, and all our tests deliver exactly the same result as on the mainframe.

Gathering specifications

- Specification? *Was ist das?*
- More often than not
 - 4GLs are pragmatic tools with no formal specification
 - Implementations of 3GLs are also “pragmatic” in that sense
- Documentation when available is not reliable
- Derived semantics
 - Generates intermediate COBOL code → inherits COBOL behavior
 - Generates intermediate Java code → ...
- Validation by testing
 - Unit testing
 - Corner cases

Redefining the very concept of language

- Different forms of input
- Technical textual representations
 - Would have been XML
 - If these tools weren't much much older than XML!

Legal issues

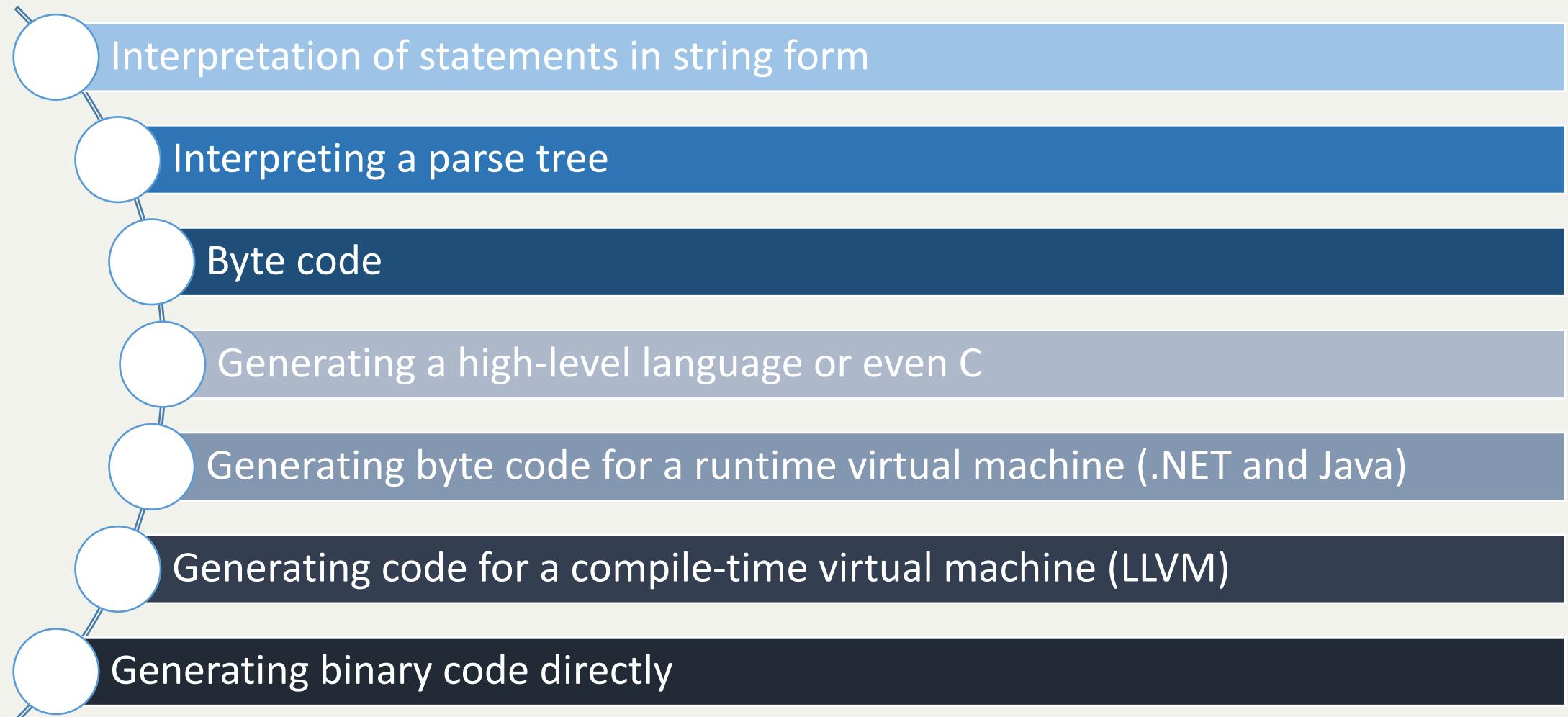
- After quite some wrangling...
 - Oracle vs. Google on Android's API's
- ... the decision has come:

A language cannot be copyrighted

https://en.wikipedia.org/wiki/SAS_Institute_Inc_v_World_Programming_Ltd

- But an implementation can
 - And so can its documentation!
- Protect yourself: never get close to any copyrighted material
 - Ask questions to someone with expertise and access to the information
 - Documented process

Redefining the very concept of a compiler...



Compiler

Compiler, IDE

Compiler, IDE, Database
connector

Compiler, IDE, Database connector,
Debugger support

Compiler, IDE, Database connector,
Debugger support, Compile-time
SQL dialect conversion

Compiler, IDE, Database connector, Debugger support, Compile-time SQL dialect conversion, Interface manager

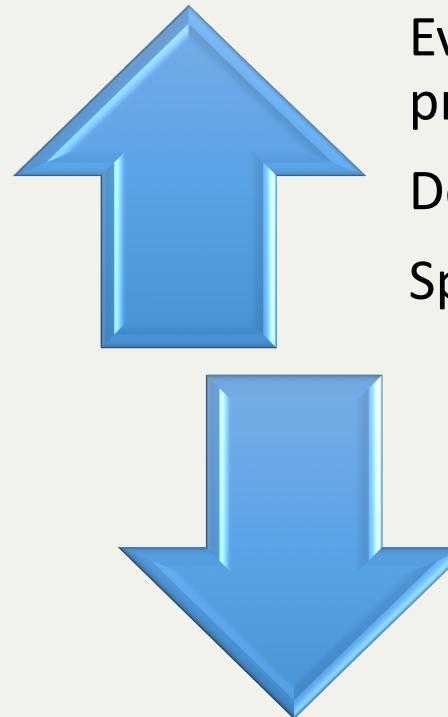
Compiler, IDE, Database connector, Debugger support, Compile-time SQL dialect conversion, Interface manager, Support infrastructure

Compiler, IDE, Database connector, Debugger support, Compile-time SQL dialect conversion, Interface manager, Support infrastructure,
Regression testing infrastructure

Compiler, IDE, Database connector, Debugger support,
Compile-time SQL dialect conversion, Interface
manager, Support infrastructure,
Regression testing infrastructure,

Documentation

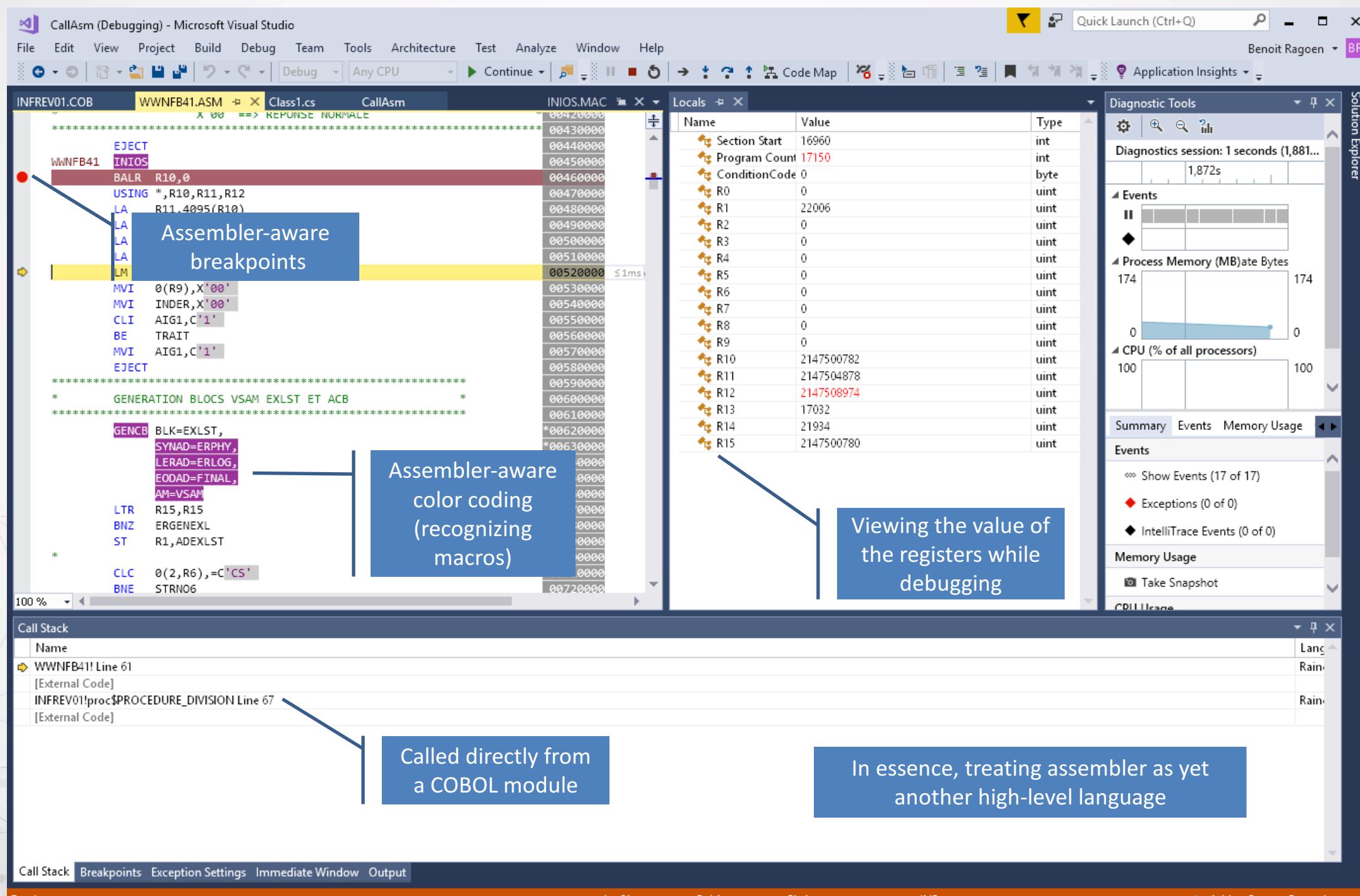
Time vs. performance



Ever increasing
processing power
Decreasing costs
Specialized processors

Increasing needs
Digitalization
Infrastructure

- Caveat: in virtual machines
 - Functional behaviours are documented
 - Performance characteristics are not (and evolving)



In conclusion...

- Compilers are here to stay
- Essential devices to resist erosion of time
- But there is much more to them than parsing/semantic analysis/code generation

D. Lexical Variations



raincode **LABS**

compiler experts

Quiz 1:

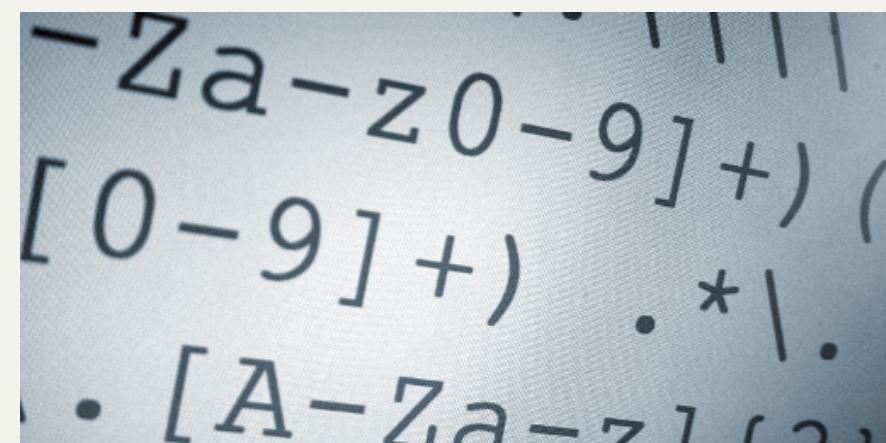
What is the worst case for
lexical analysis based on Unix's
lex ?

$A \rightarrow a^*b$
 $\rightarrow a$

Input: a^n

Quiz 2:

Right to left lexical analysis?





Theoretical reversibility does
not imply equivalent
performances

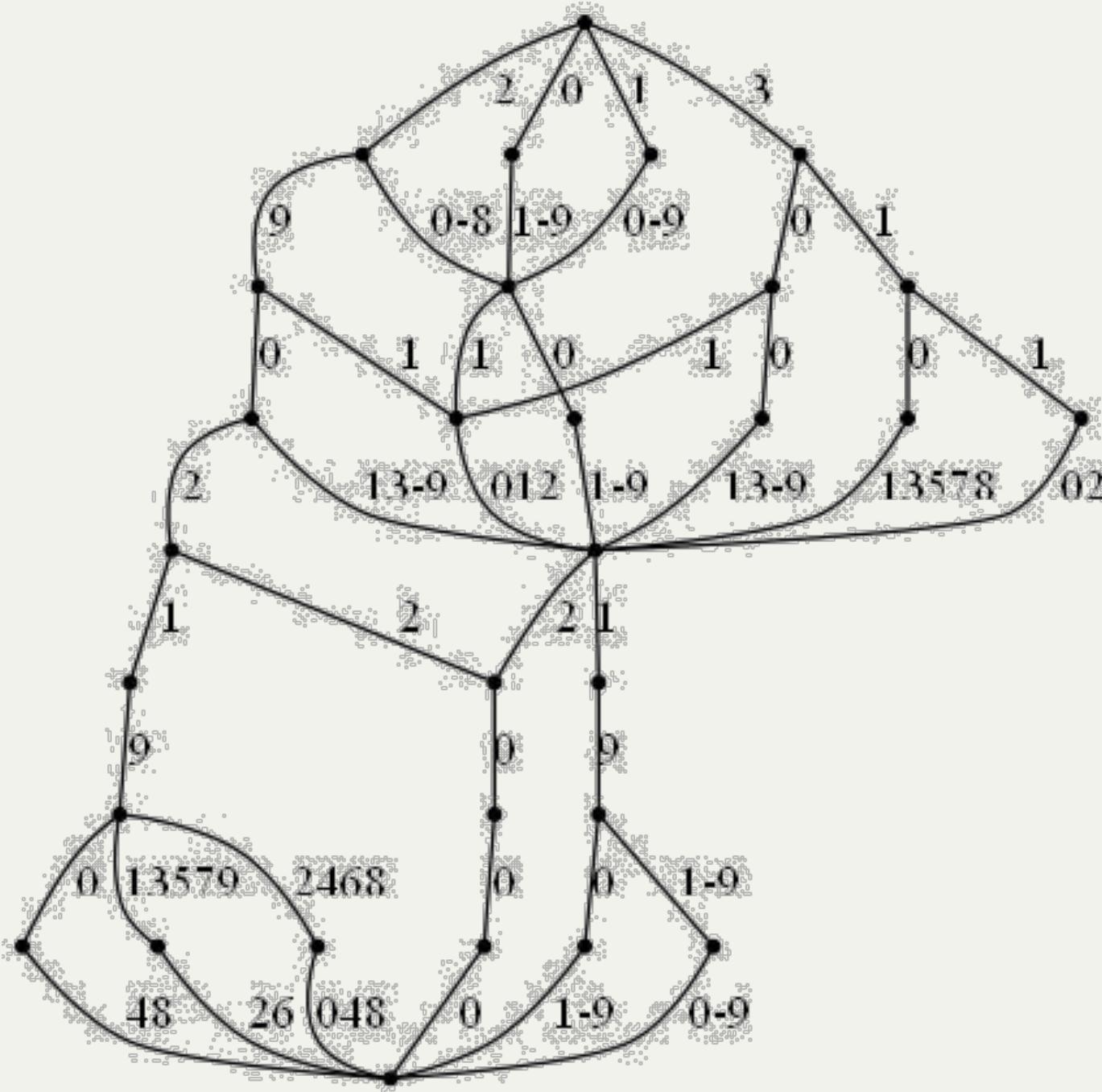
Quiz 3:

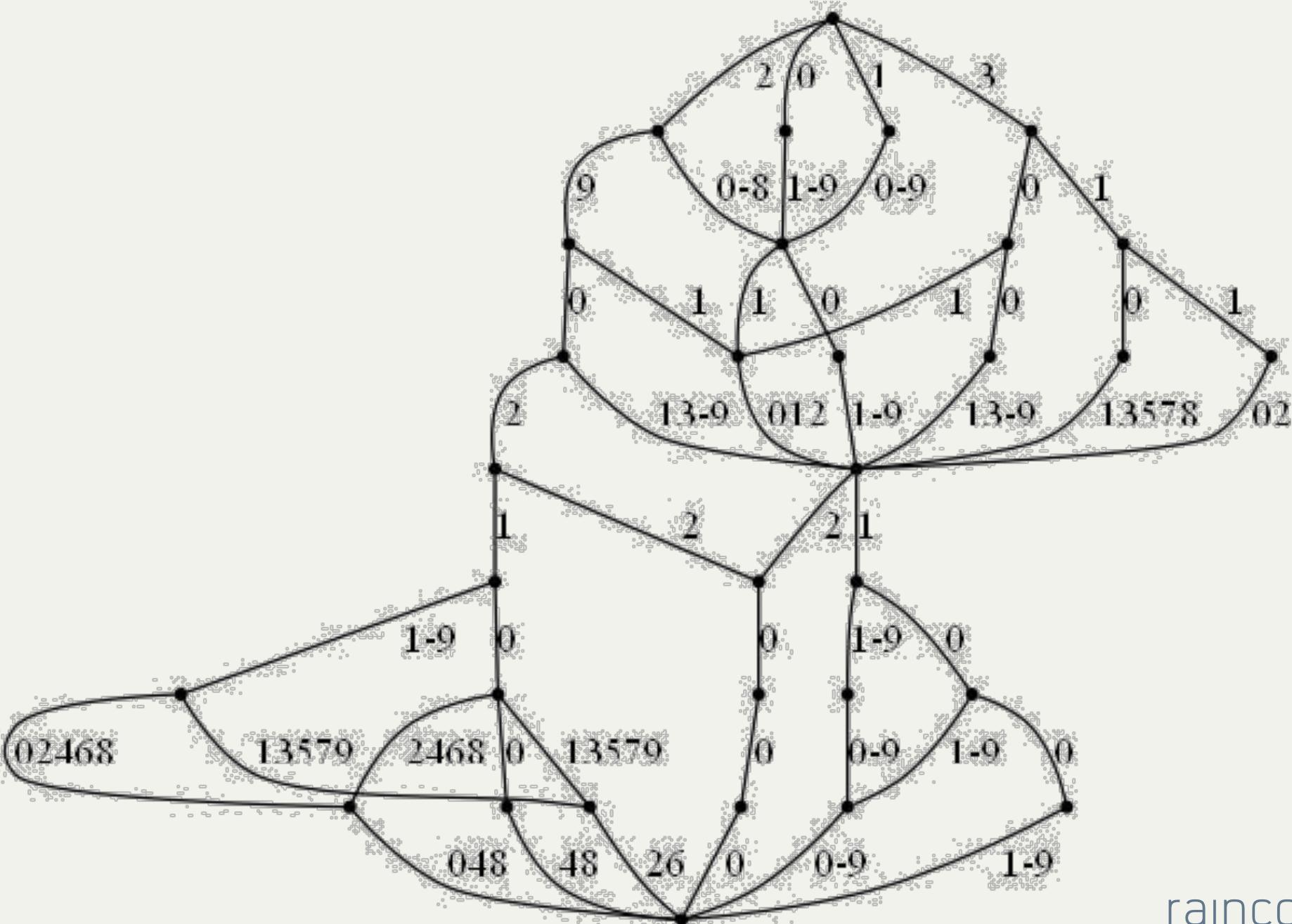
Why are pumping lemmas
important?

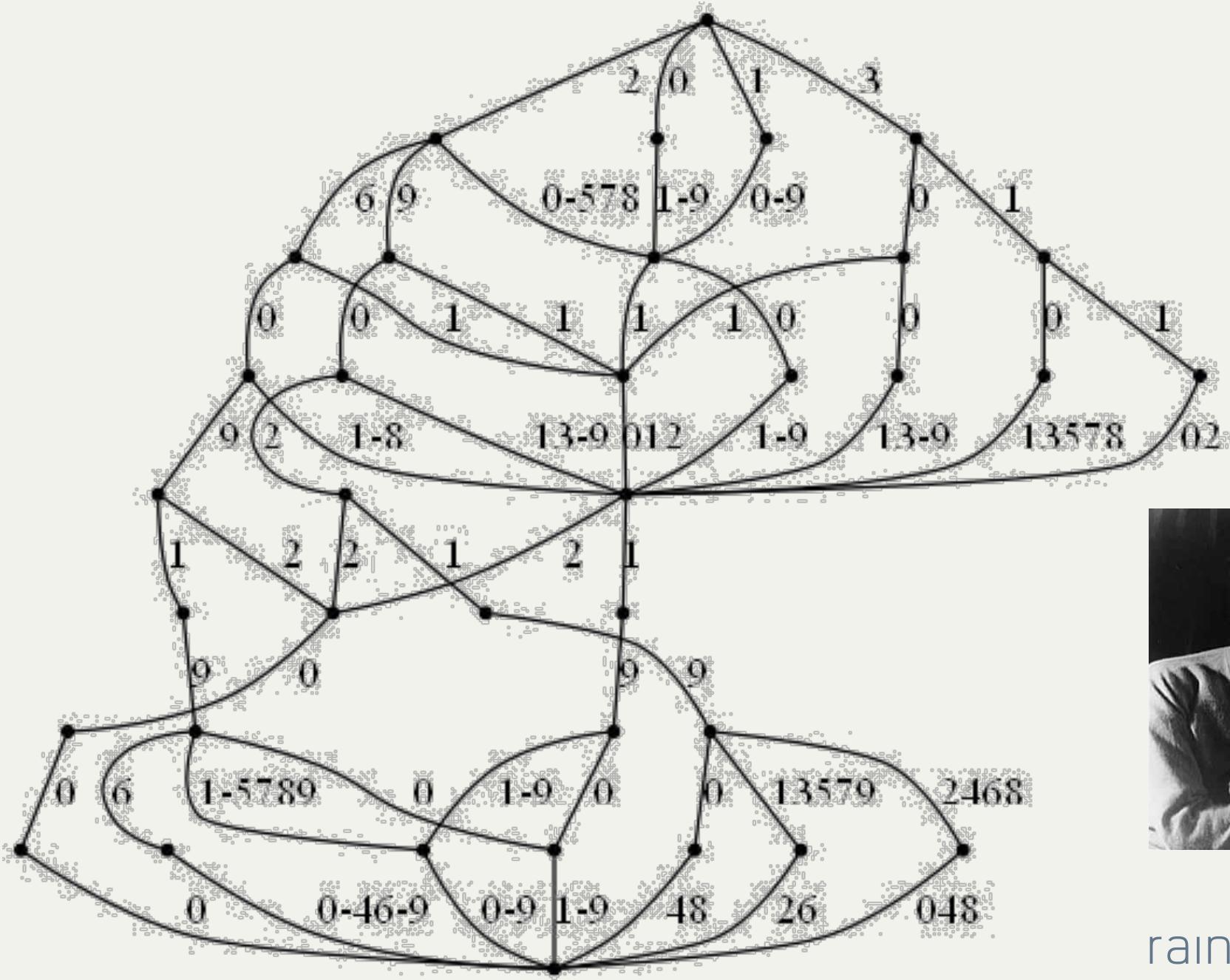


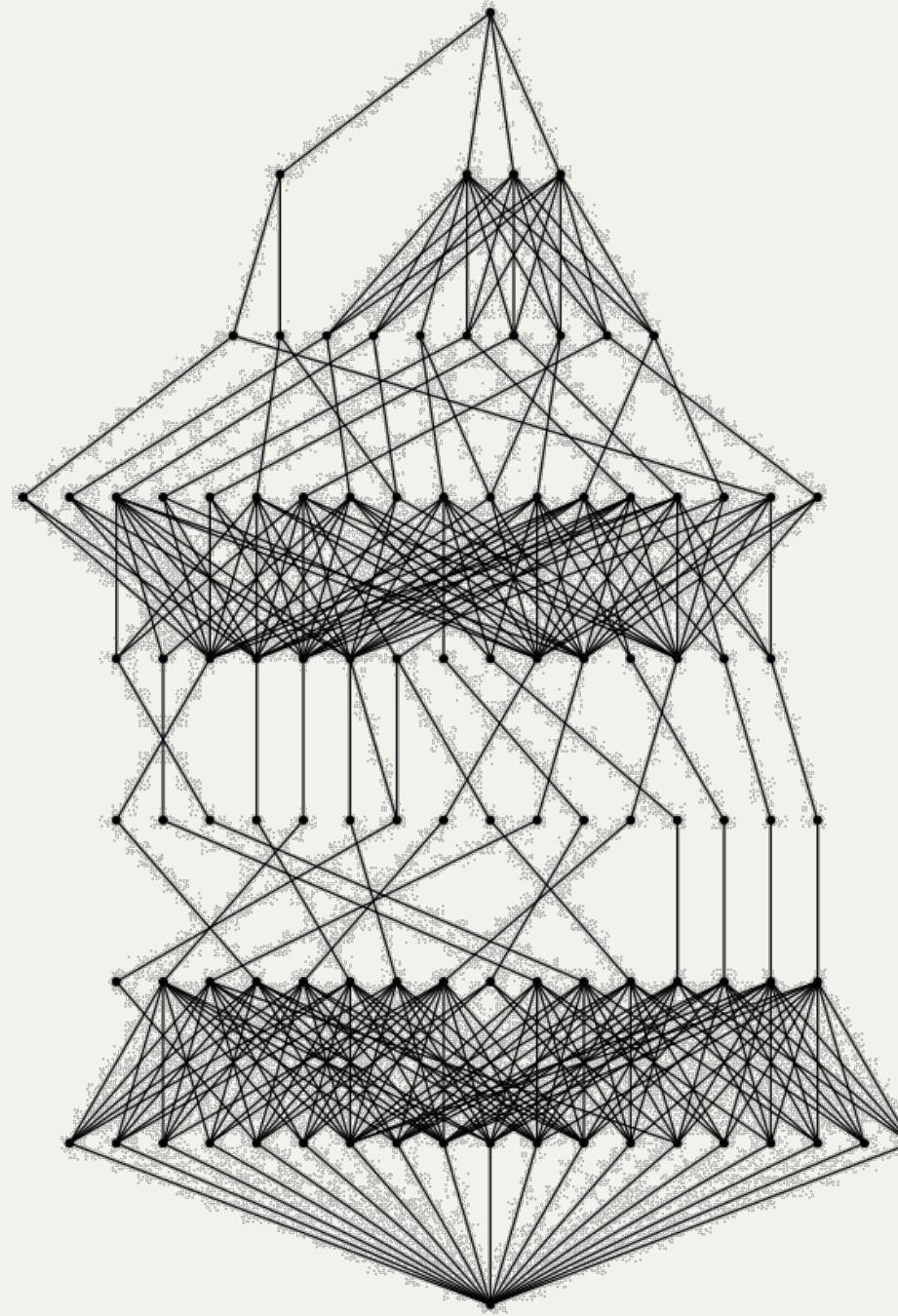
Quiz 4:

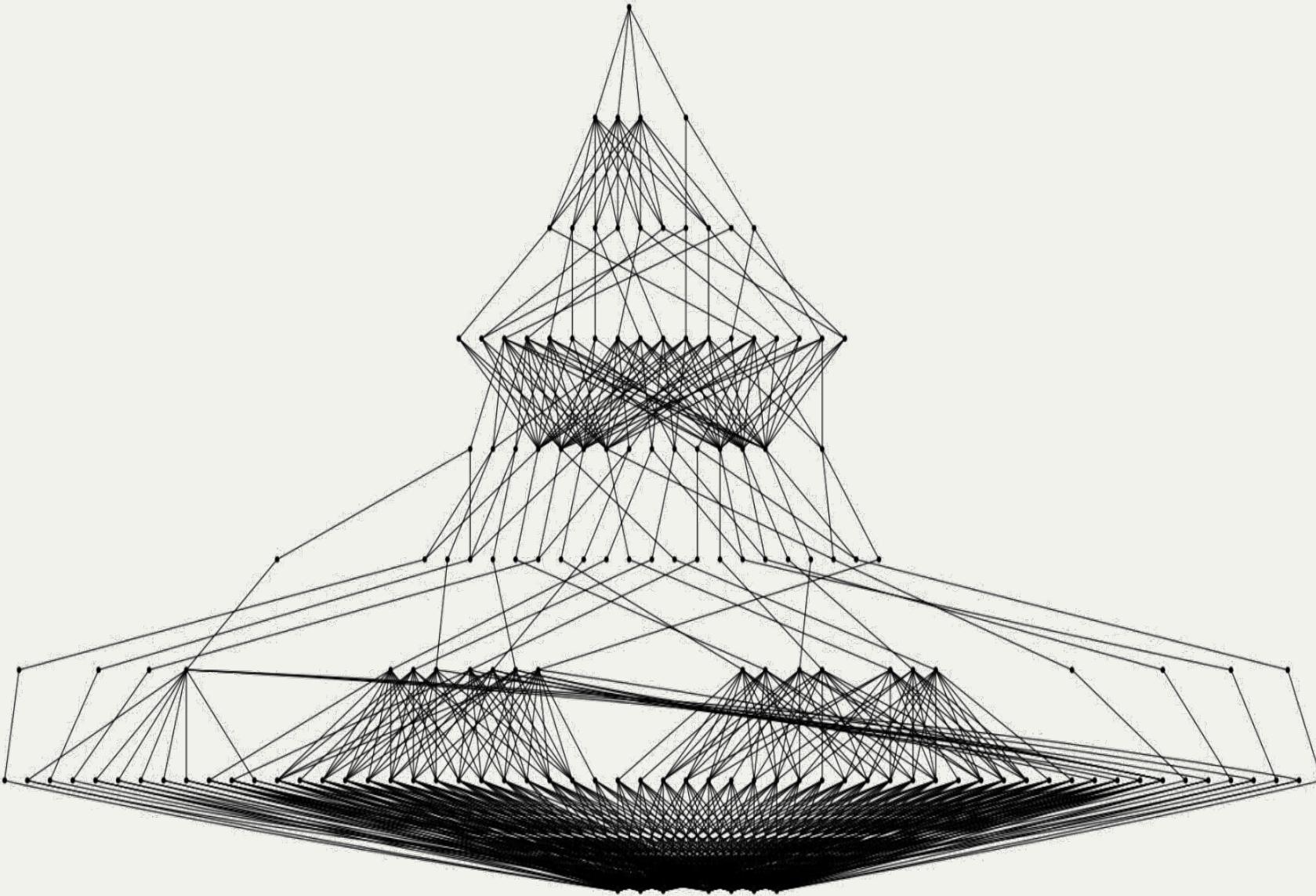
How to check for a valid date of
the 20th or 21st century?











raincode **LABS**
compiler experts

((0[1-9])|(1[0-9]))((0[1-9])|(1[012]))((19((0[1-9])|([1-9][0-9])))|(2000))|(2((9((0((2((19((0[48])|([13579][26])|([2468][048])))|(2000))|[([13-9]((19((0[1-9])|([1-9][0-9])))|(2000))))|(1[012]((19((0[1-9])|([1-9][0-9])))|(2000))))|(([0-8]((0[1-9])|(1[012])))((19((0[1-9])|([1-9][0-9])))|(2000))))|(3((0((0[13-9])|(1[012])))|(1((0[13578])|(1[02]))))|(19((0[1-9])|([1-9][0-9])))|(2000)))



Theoretical equivalence
is just that.

Theoretical.

Contribution: lexical conjunction

(abc?)⁺&.{35-40}

That's all folks!!!



RAINCODE HEADQUARTERS



45 Rue de la Caserne
B-1000 Brussels, Belgium



+32 2 522 06 63



info@raincodelabs.com

RAINCODE USA



13245 Atlantic Boulevard. Suite 4-263
Jacksonville, FL 32225, USA



+1 412.552.8207



info@raincodelabs.com

www.raincodelabs.com

raincode LABS

compiler experts