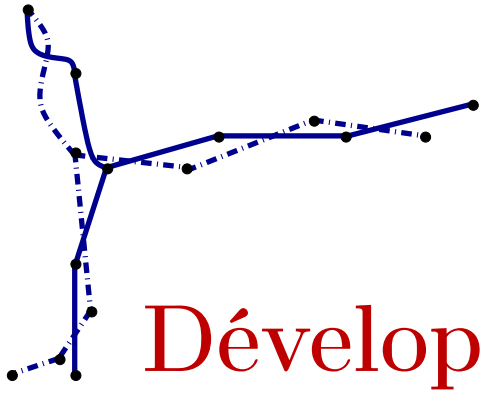
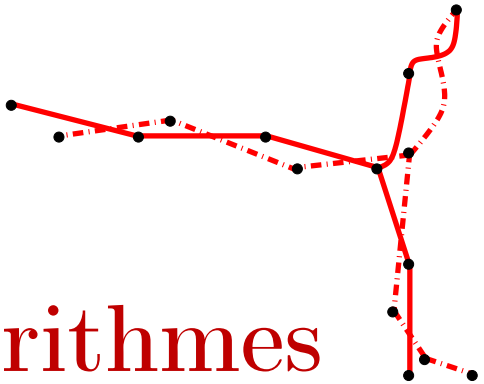
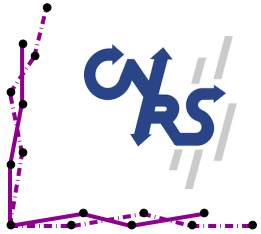


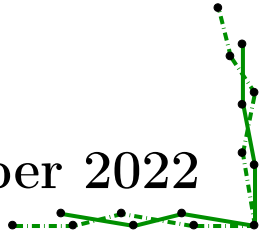


Développement des Algorithmes d'Application Réticulaire

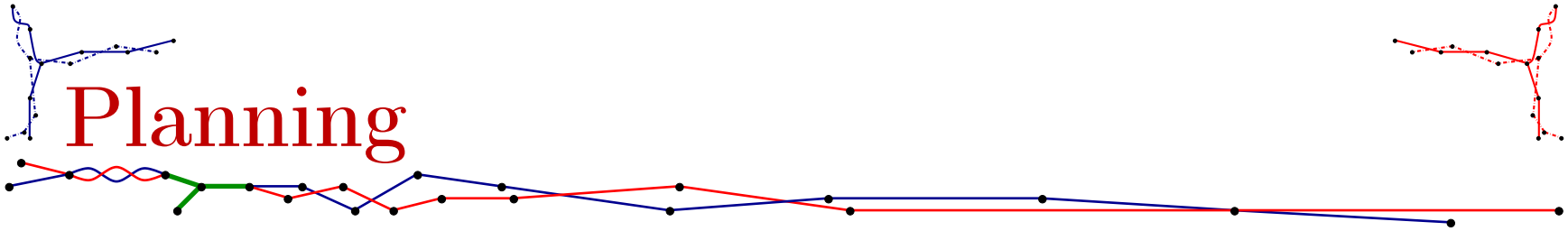
https://www-apr.lip6.fr/~buixuan/daar2022

Binh-Minh Bui-Xuan





PARIS, September 2022



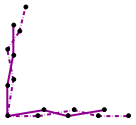
Planning

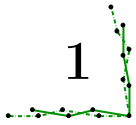
Teaching team :

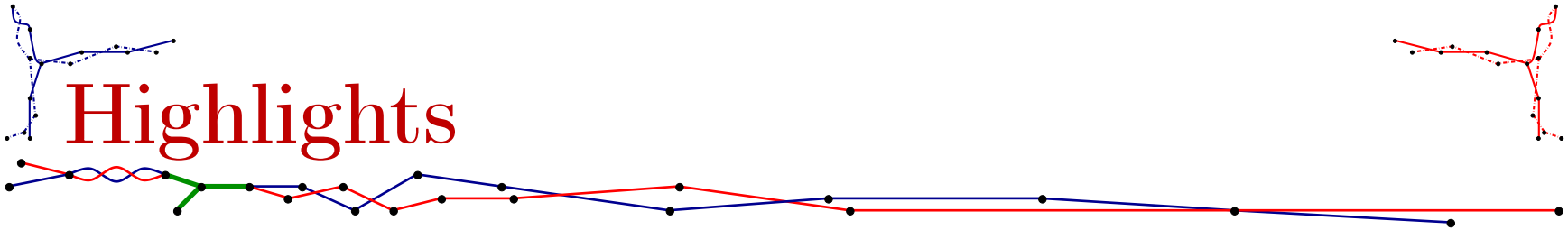
- Binh-Minh Bui-Xuan (lectures)
- Arthur Escriou (lectures 2, 13, 14 + TME)
- Alfred Deivassayagame (lectures 4, 5 + TME)
- Guillaume Hivert (lectures 6, 7 + TME)
- Suxue Li (TME)

Assessment :

- session 1 = projects (70% total) + written exam (30%)
- session 2 = written exam (100%)







Highlights

Search engine :

– desktop algorithm : RegEx ; KMP ; P-tries

– graph algorithm : data eng. (ES) ; decentralized projects (ETH) ; stream data...

Project teams :

– lectures 1-7 : backend algorithms

– lectures 8-14 : graph algorithms + frontend development

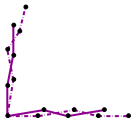
– final project : many choices !

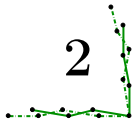
librarian app (choice A, backend)

research project (choice B, routing algorithms)

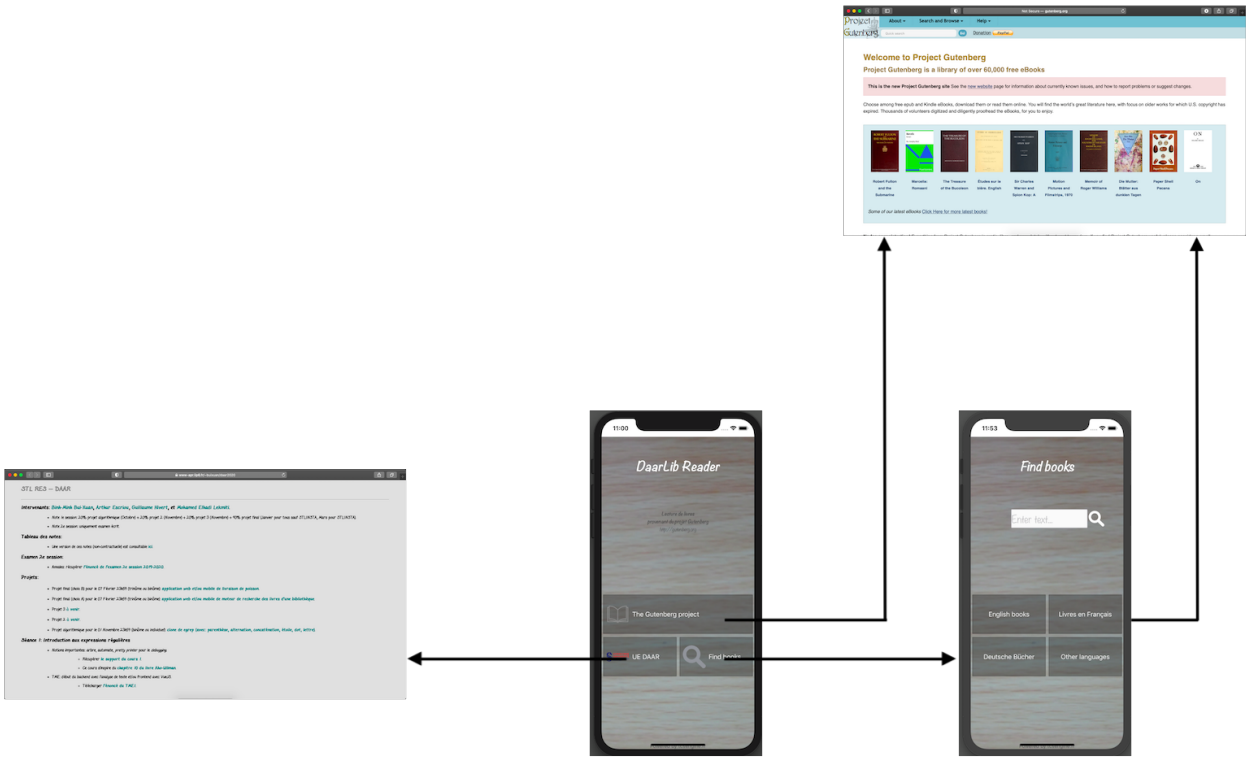
itinerary app (choice C, backend++)

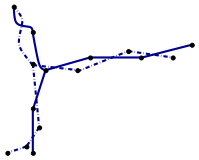
...

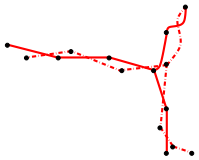




Final project example – books

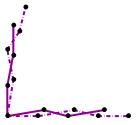


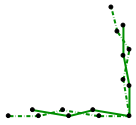


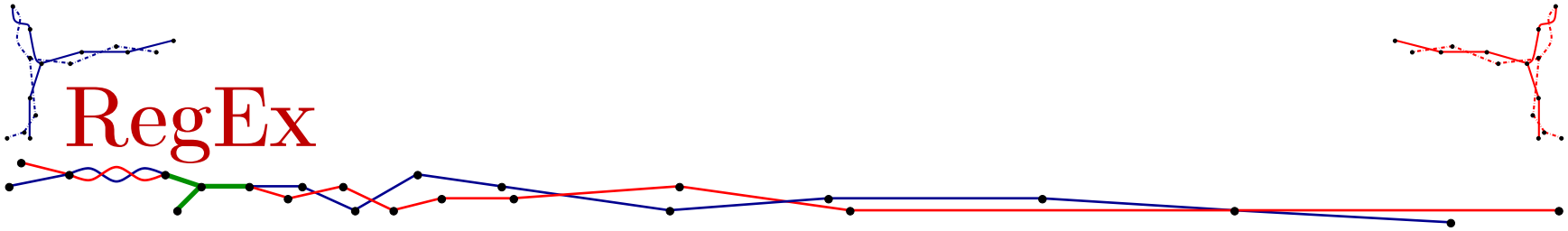


Search engine :

desktop algorithm







REGEX (INPUT) $\rightarrow$ AUTOMATON (OUTPUT)

$$\begin{aligned} \text{RegEx} &\rightarrow \text{RegEx with parenthesis (RegEx tree)} && (1)\\ &\rightarrow \text{automaton with } \epsilon - \text{transitions} && (2)\\ &\rightarrow \text{deterministic finite automaton (DFA)} && (3)\\ &\rightarrow \text{idem, minimum (optional step)} && (4) \end{aligned}$$

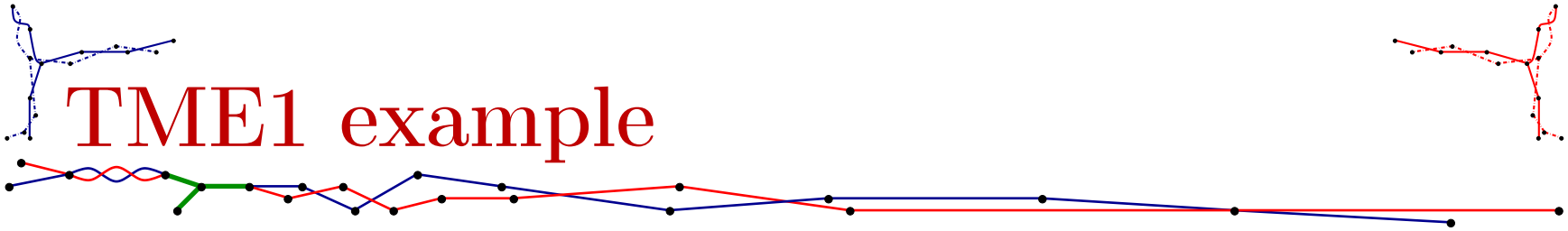
PATTERN RECOGNITION :

$$\text{text input} + \text{DFA} \rightarrow \text{boolean} \qquad (5)$$

Aho-Ullman book, ex. 10.25, p. 576

Principal steps :

RegEx "a|bc*", RegEx tree in Figure 10.29	(1)
Figure 10.30-33 (recursive calls)	(2)
Determinisation Figure 10.35	(3)
Minimisation Figure 10.36	(4)
Recognition : text input "babybcc"	(5)

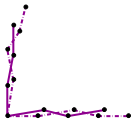


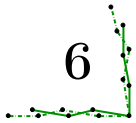
TME1 example

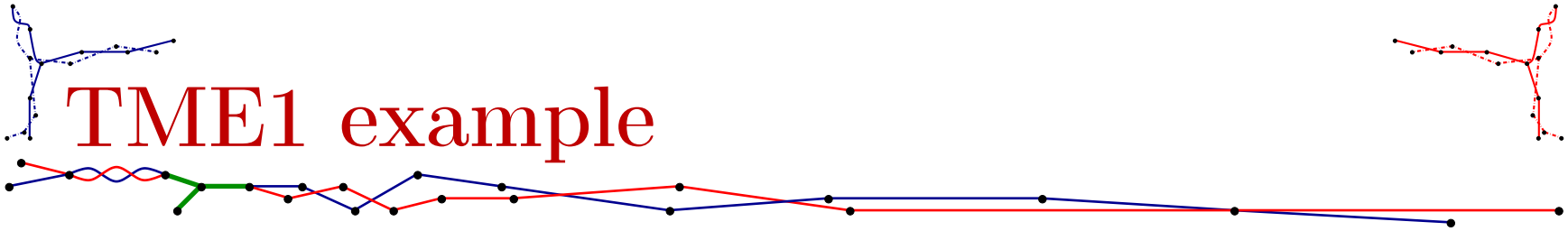
Principal steps :

RegEx "S(a|g|r)+on", RegEx tree ?	(1)
Figure 10.27 + 10.28 (recursive calls) ?	(2)
Determinisation (with subset algorithm) ?	(3)
Naive minimisation p.555 ?	(4)
Recognition : text input Babylonia history	(5)

N.B. : for project 1, we restrict ourselves to : parenthesis, alternation, concatenation, star, dot, ASCII letter.







TME1 example

Principal steps :

RegEx "S(a|g|r)+on", RegEx tree ? (1)

Figure 10.27 + 10.28 (recursive calls) ? (2)

Determinisation (with subset algorithm) ? (3)

Naive minimisation p.555 ? (4)

Recognition : text input Babylonia history (5)

Next step : project 1, instructions on

https://www-apr.lip6.fr/~buixuan/daar2022