Balanced ω-regular languages

Luc Edixhoven

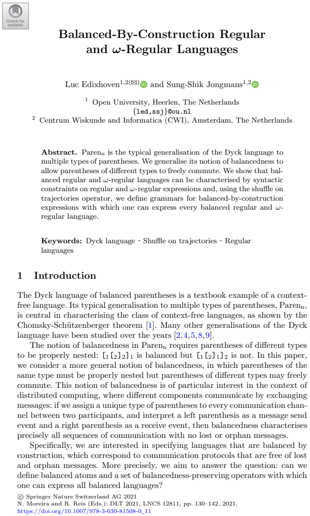
19 April 2022

Who am I?

Internal PhD student since september 2019

Under the enlightening supervision of Sung(-Shik Jongmans)

Why this talk?



Check for updates

Balanced-By-Construction Regular and ω-Regular Languages

Luc Edixhoven[1,2](✉) and Sung-Shik Jongmans[1,2]

[1] Open University, Heerlen, The Netherlands
{led,ssj}@ou.nl
[2] Centrum Wiskunde and Informatica (CWI), Amsterdam, The Netherlands

Abstract. Paren_n is the typical generalisation of the Dyck language to multiple types of parentheses. We generalise its notion of balancedness to allow parentheses of different types to freely commute. We show that balanced regular and ω-regular languages can be characterised by syntactic constraints on regular and ω-regular expressions and, using the shuffle on trajectories operator, we define grammars for balanced-by-construction expressions with which one can express every balanced regular and ω-regular language.

Keywords: Dyck language · Shuffle on trajectories · Regular languages

1 Introduction

The Dyck language of balanced parentheses is a textbook example of a context-free language. Its typical generalisation to multiple types of parentheses, Paren_n, is central in characterising the class of context-free languages, as shown by the Chomsky-Schützenberger theorem [1]. Many other generalisations of the Dyck language have been studied over the years [2,4,5,8,9].

The notion of balancedness in Paren_n requires parentheses of different types to be properly nested: $[_1[_2]_2]_1$ is balanced but $[_1[_2]_1]_2$ is not. In this paper, we consider a more general notion of balancedness, in which parentheses of the same type must be properly nested but parentheses of different types may freely commute. This notion of balancedness is of particular interest in the context of distributed computing, where different components communicate by exchanging messages: if we assign a unique type of parentheses to every communication channel between two participants, and interpret a left parenthesis as a message send event and a right parenthesis as a receive event, then balancedness characterises precisely all sequences of communication with no lost or orphan messages.

Specifically, we are interested in specifying languages that are balanced by construction, which correspond to communication protocols that are free of lost and orphan messages. More precisely, we aim to answer the question: can we define balanced atoms and a set of balancedness-preserving operators with which one can express all balanced languages?

N. Moreira and R. Reis (Eds.): DLT 2021, LNCS 12811, pp. 130–142, 2021.
https://doi.org/10.1007/978-3-030-81508-0_11

Disclaimer

The announcement:

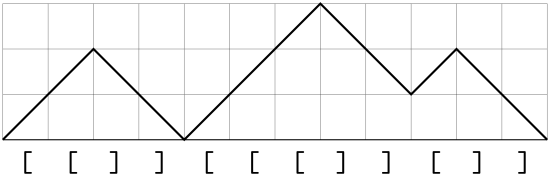
Disclaimer

The announcement:

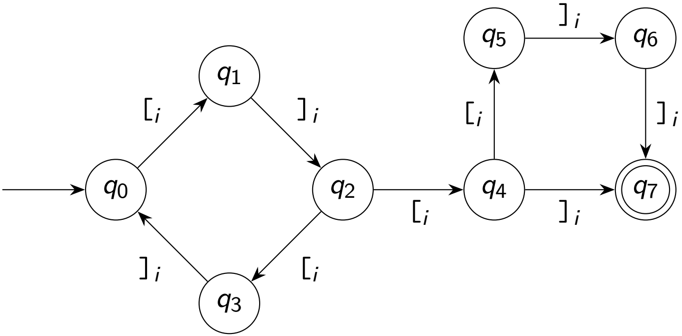
The actual talk:

What was the first half about?

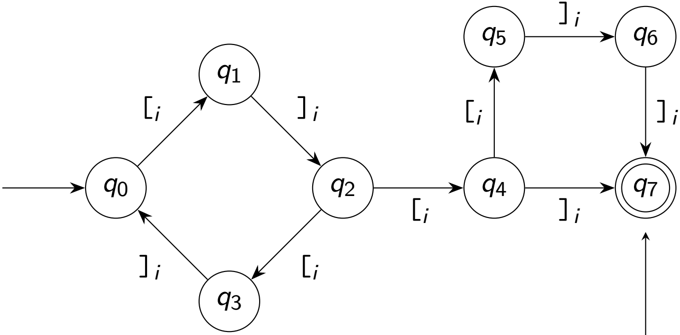
- Formal languages over brackets $[_1,]_1, [_2,]_2, \ldots$
- Words are *balanced* if all brackets occur in ordered pairs
  - $[_1]_1 [_2]_2$ is balanced
  - $[_1 [_2]_1]_2$ is balanced (interleaving is fine)
  - $[_1 [_2]_1$ is not, nor is $]_1 [_2 [_1]_2$
- A language is balanced if all of its words are, and so are automata and expressions



Balanced automata

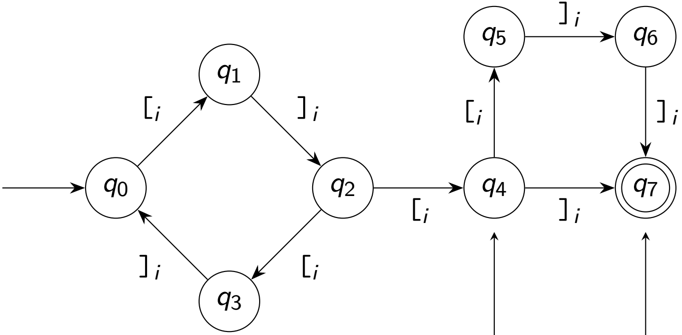


Balanced automata



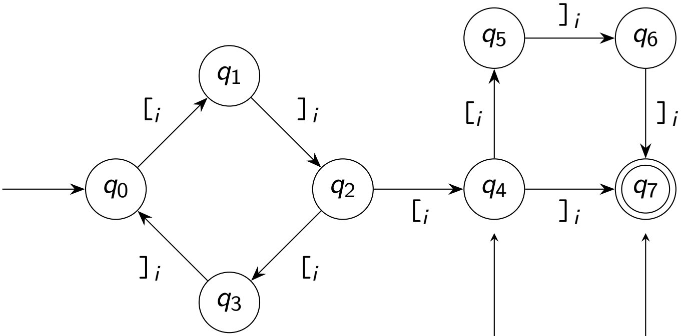
- q_7 must have an *i-balance* of 0

Balanced automata



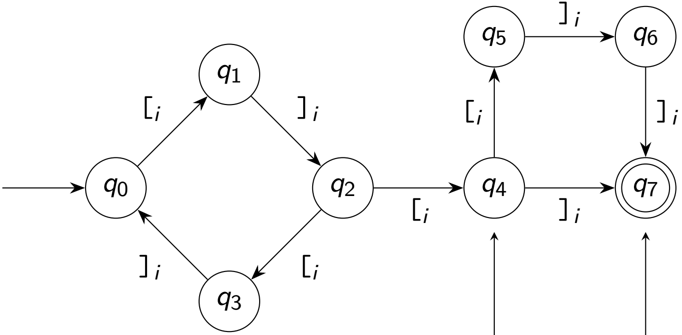
- q_7 must have an i-*balance* of 0
- q_4 must have an i-balance of 1

Balanced automata



- q_7 must have an *i-balance* of 0
- q_4 must have an i-balance of 1
- ...

Balanced automata



- q_7 must have an i-*balance* of 0
- q_4 must have an i-balance of 1
- ...
- All balances must be non-negative, initial state must be 0

Balanced regular expressions

- $[_i$ has an i-balance of 1, $]_i$ has an i-balance of -1
- $[_i]_i$ has an i-balance of 0...

Balanced regular expressions

- $[_i$ has an i-balance of 1, $]_i$ has an i-balance of -1
- $[_i]_i$ has an i-balance of 0... but so does $]_i[_i$
- Solution: *minimum i-balance*. That of $[_i]_i$ is 0, while that of $]_i[_i$ is -1.

Balanced regular expressions

- $\texttt{[}_i$ has an i-balance of 1, $\texttt{]}_i$ has an i-balance of -1
- $\texttt{[}_i\texttt{]}_i$ has an i-balance of 0... but so does $\texttt{]}_i\texttt{[}_i$
- Solution: *minimum i-balance*. That of $\texttt{[}_i\texttt{]}_i$ is 0, while that of $\texttt{]}_i\texttt{[}_i$ is -1.

- The i-balance of $\texttt{[}_i + (\texttt{[}_i\texttt{]}_i\texttt{[}_i)$ is 1
- The i-balance of $\texttt{[}_i + (\texttt{[}_i\texttt{]}_i)$ is undefined
- The i-balance of $(\texttt{[}_i\texttt{]}_i)^*$ is 0
- The i-balance of $(\texttt{[}_i)^*$ is undefined

Balanced regular expressions

- $\texttt{[}_i$ has an i-balance of 1, $\texttt{]}_i$ has an i-balance of -1
- $\texttt{[}_i\texttt{]}_i$ has an i-balance of 0... but so does $\texttt{]}_i\texttt{[}_i$
- Solution: *minimum i-balance*. That of $\texttt{[}_i\texttt{]}_i$ is 0, while that of $\texttt{]}_i\texttt{[}_i$ is -1.

- The i-balance of $\texttt{[}_i + (\texttt{[}_i\texttt{]}_i\texttt{[}_i)$ is 1
- The i-balance of $\texttt{[}_i + (\texttt{[}_i\texttt{]}_i)$ is undefined
- The i-balance of $(\texttt{[}_i\texttt{]}_i)^*$ is 0
- The i-balance of $(\texttt{[}_i)^*$ is undefined

Balances and minimum balances should all be 0.

Balanced expression grammar

$$e ::= \varnothing \mid \lambda \mid [_1 \mid]_1 \mid [_2 \mid]_2 \mid \dots \\ \mid e \cdot e \mid e + e \mid e^*$$

Can express all balanced regular languages, but also unbalanced ones.

Balanced expression grammar

$$\begin{aligned} e ::= {} & \varnothing \mid \lambda \mid [_1 \mid]_1 \mid [_2 \mid]_2 \mid \ldots \\ & \mid e \cdot e \mid e + e \mid e^* \end{aligned}$$

Can express all balanced regular languages, but also unbalanced ones.

Solution:

$$\begin{aligned} e ::= {} & \varnothing \mid \lambda \mid [_1 \cdot]_1 \mid [_2 \cdot]_2 \mid \ldots \\ & \mid e_1 \cdot e_2 \mid e_1 + e_2 \mid e^* \\ & \mid \sqcup^1_\theta(e_1) \mid \sqcup^2_\theta(e_1, e_2) \mid \ldots \end{aligned}$$

$$\begin{aligned} \theta ::= {} & \varnothing \mid \lambda \mid 1 \mid 2 \mid \ldots \\ & \mid \theta_1 \cdot \theta_2 \mid \theta_1 + \theta_2 \mid \theta^* \end{aligned}$$

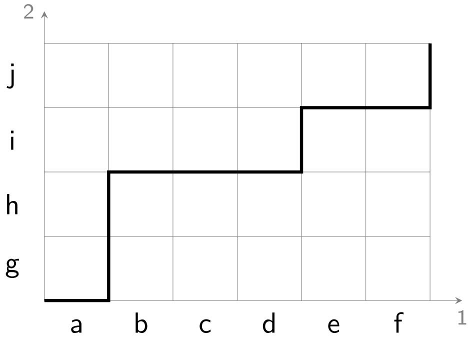
Shuffle on trajectories

Mateescu et al.: "Shuffle on trajectories" (1998)

Trajectory

$\sqcup\!\sqcup^{2}_{1221112112}(\text{abcdef}, \text{ghij})$

$= \text{aghbcdiefj}$



- Only defined if the trajectory fits the operands
- Generalises to languages and expressions

Soundness

"All expressions are balanced and regular"

- Proof by constructing a balanced automaton

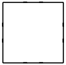
Completeness

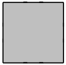
"All balanced regular languages can be expressed"

- Rewrite regular expression in normal form to get rid of $+$
- Rewrite subexpressions as shuffles of 'factors'
- Number of unbalanced factors correlates with balance and minimum balance

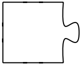
Factors

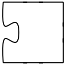
Balanced factors

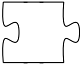
$\langle\;\rangle_i^k = ([_i]_i)^k([_i]_i)^* \rightarrow$ 

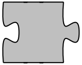
$\langle\lambda\rangle_i^k = (\langle\;\rangle_i^k)^* \rightarrow$ 

Unbalanced factors

$\langle+\rangle_i^k = \langle\;\rangle_i^k [_i \rightarrow$ 

$\langle-\rangle_i^k =]_i \langle\;\rangle_i^k \rightarrow$ 

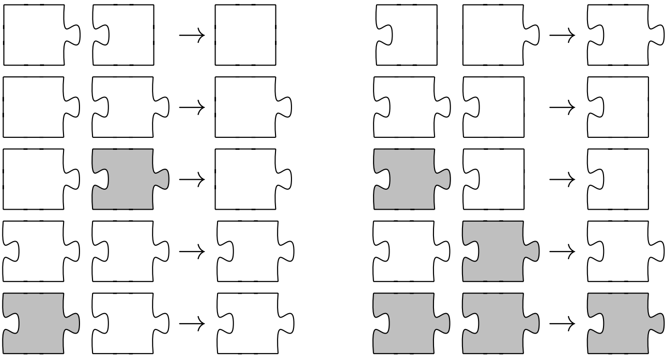
$\langle\pm\rangle_i^k =]_i \langle\;\rangle_i^k [_i \rightarrow$ 

$\langle\star\rangle_i^k = (\langle\pm\rangle_i^k)^* \rightarrow$ 

Lemma (Merge)

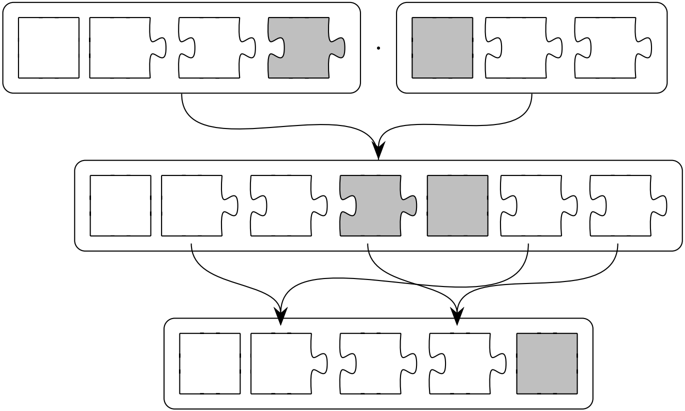
If [. . .] then $\bigsqcup_{T}(L_1, \ldots, L_{n-1}, L_n) = \bigsqcup_{T'}(L_1, \ldots, L_{n-1}L_n)$.

Specifically:



Lemma (Concatenation)

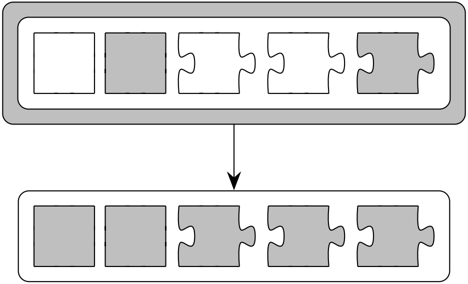
If [...] then $\sqcup_{T_1}(L_1, \ldots, L_n) \cdot \sqcup_{T_2}(L_{n+1}, \ldots, L_{n+m}) = \sqcup_T(L_1, \ldots, L_{n+m}) = \sqcup_{T'}(L_k, \ldots, L_\ell)$.



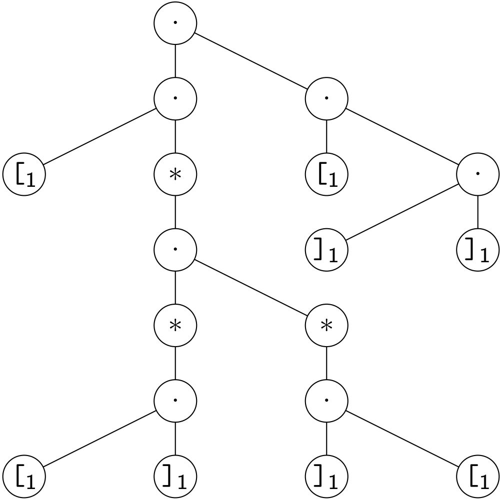
Loops

Lemma (Star)

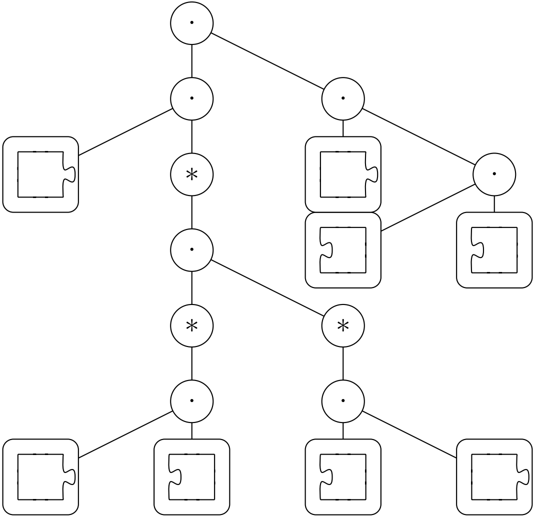
If [...] then $(\sqcup_T(L_1,\ldots,L_n))^* = \sqcup_{T^*}(L_1^*,\ldots,L_n^*)$.



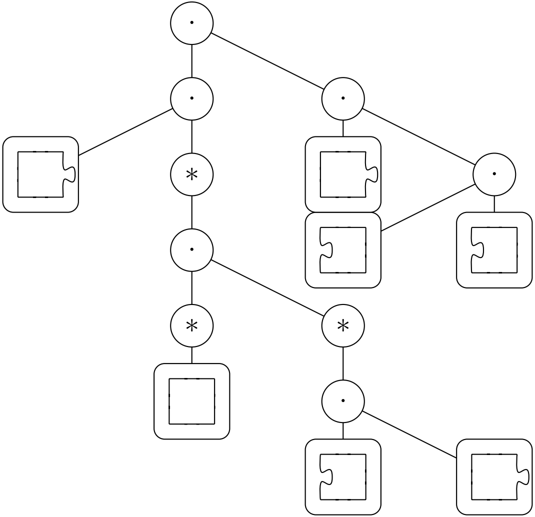
Example



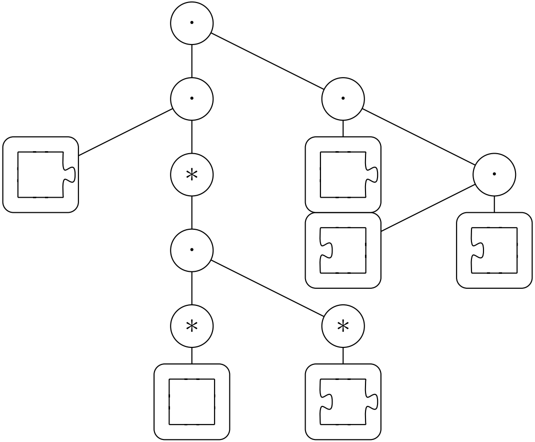
Example



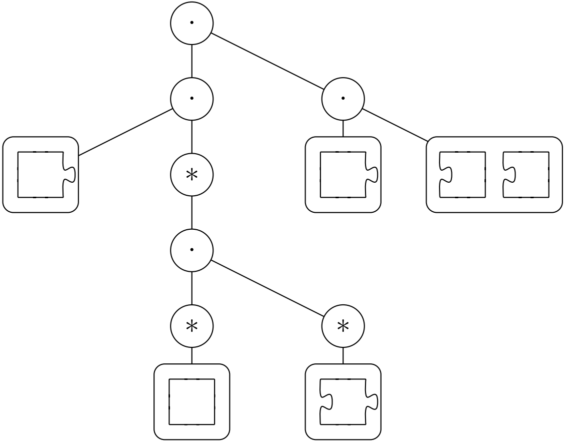
Example



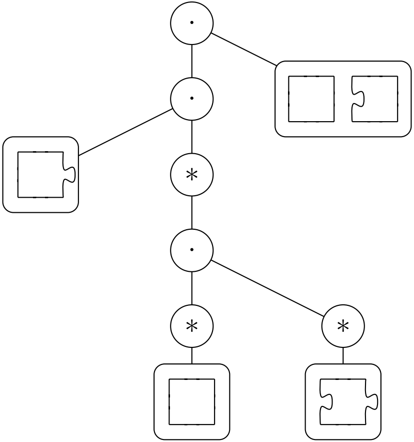
Example



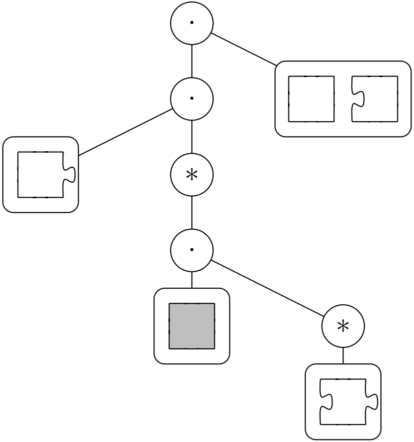
Example



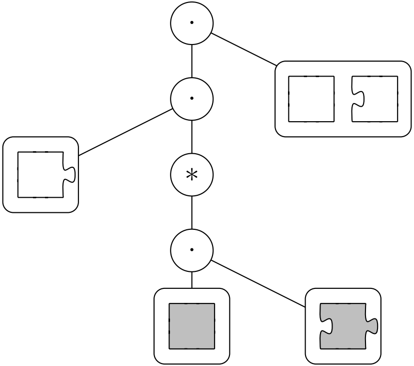
Example



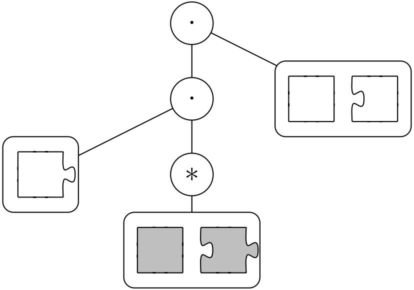
Example



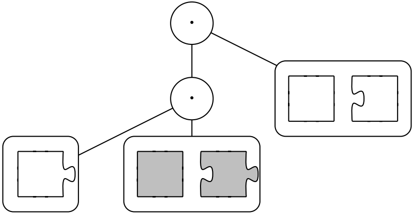
Example



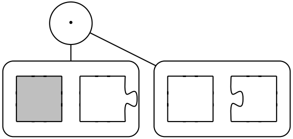
Example



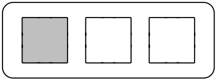
Example



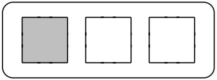
Example



Example

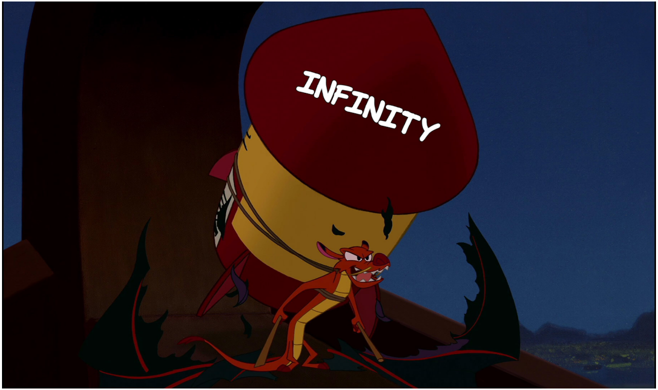


Example



What's new?

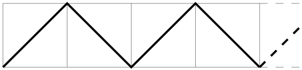
What's new?



Balancedness

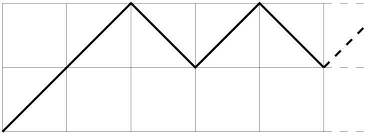
Balancedness += Boundedness

$(\texttt{[]})^{\omega}$



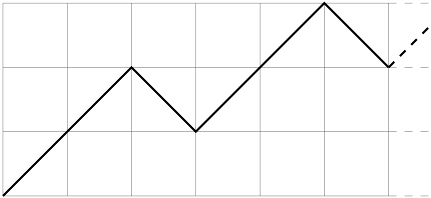
✓

$\texttt{[}(\texttt{[]})^{\omega}$



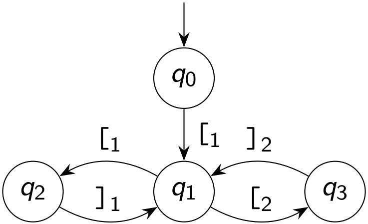
✓

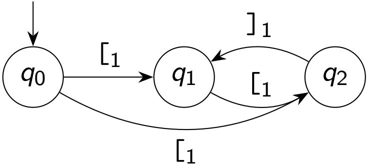
$(\texttt{[[]})^{\omega}$



✗

Balanced ω-automata





$\{\{q_1, q_2\}, \{q_1, q_3\}, \{q_1, q_2, q_3\}\}$

$[_1([_1]_1 + [_2]_2)^\omega$

Unbalanced

Modified acceptance criterion

$\{\{q_1, q_2\}\}$

$(\lambda + [_1)([_1]_1)^\omega$

Balanced

Split balance in two

Balanced ω-regular expressions

As before, but:

- Keep track of guaranteed infinite ocurrences
- Balance now consists of both a lower and an upper bound

- $[_1([_1]_1 + [_2]_2)^\omega$: 2-balance is between 0 and 0, 1-balance is between 1 and 1; 1-brackets not guaranteed to occur infinitely often
- $(\lambda + [_1)([_1]_1)^\omega$: 1-balance is 0; 1-brackets guaranteed to occur infinitely often

Balance bounds and minimum balances should all be 0.

Balanced ω-expression grammar

$$
\begin{aligned}
e &::= \varnothing \mid e + e \mid E \cdot e \mid E_+^\omega \mid \text{Ш}_{T_\omega}(C, \ldots, C) && (\omega\text{-regular}) \\
E &::= \varnothing \mid \lambda \mid P \mid E + E \mid E \cdot E \mid E^* \mid \text{Ш}_{T}(E, \ldots, E) && (\text{regular}) \\
E_+ &::= \varnothing \mid P \mid E_+ + E_+ \mid E \cdot E_+ \cdot E \mid \text{Ш}_{T_+}(E, \ldots, E) && (\text{no } \lambda) \\
P &::= [_1 \cdot]_1 \mid [_2 \cdot]_2 \mid \ldots && (\text{parentheses}) \\
C &::= e \mid E && (\omega\text{-shuffle operand})
\end{aligned}
$$

$$
\begin{aligned}
T &::= \varnothing \mid \lambda \mid 1 \mid 2 \mid \ldots \mid T + T \mid T \cdot T \mid T^* && (\text{trajectory}) \\
T_+ &::= \varnothing \mid 1 \mid 2 \mid \ldots \mid T_+ + T_+ \mid T \cdot T_+ \cdot T && (\text{no } \lambda) \\
T_\omega &::= \varnothing \mid T_\omega + T_\omega \mid T \cdot T_\omega \mid T_+^\omega && (\omega\text{-trajectory})
\end{aligned}
$$

Soundness

"All expressions are balanced and ω-regular"

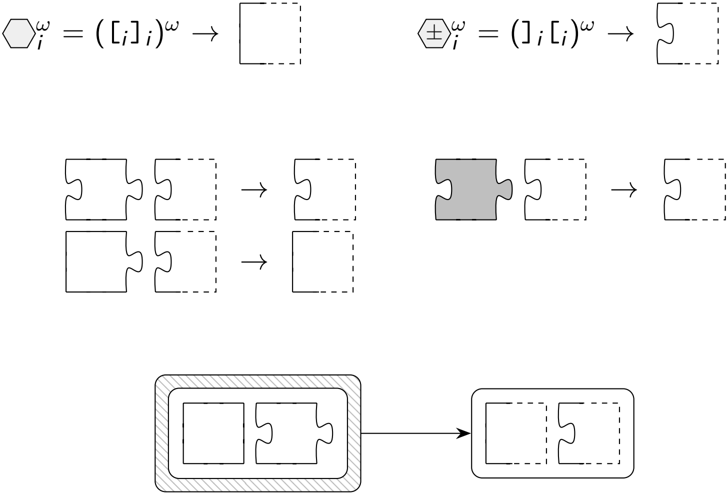
- Proof by constructing a balanced ω-automaton

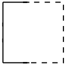
Completeness

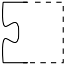
"All balanced ω-regular languages can be expressed"

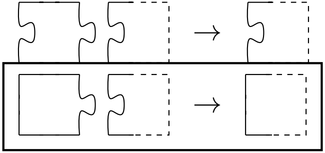
- Normal form: $e_1 e_2^\omega + \ldots + e_{2m-1} e_{2m}^\omega$
- Rewrite all e_i as shuffles of factors, then merge

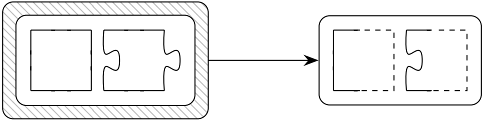
ω-factors



$_i^\omega = (\mathtt{[}_i\mathtt{]}_i)^\omega \rightarrow$ 

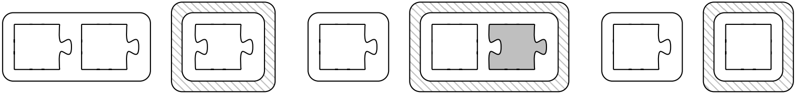
$_i^\omega = (\mathtt{]}_i\mathtt{[}_i)^\omega \rightarrow$ 

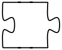
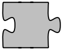
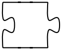
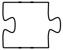




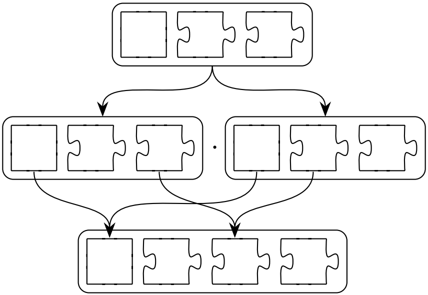
Problems

$\mathtt{[}_i\,\mathtt{[}_i(\mathtt{]}_i\,\mathtt{[}_i)^\omega$ $\quad$ $\mathtt{[}_i(\,\mathtt{[}_i\mathtt{]}_i(\mathtt{]}_i\,\mathtt{[}_i)^*)^\omega$ $\quad$ $\mathtt{[}_i(\,\mathtt{[}_i\mathtt{]}_i)^\omega$

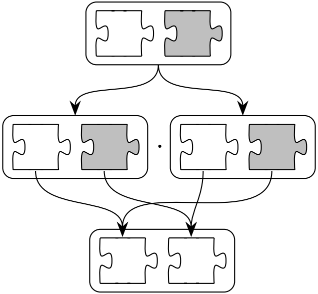


Too few   instead of  No 

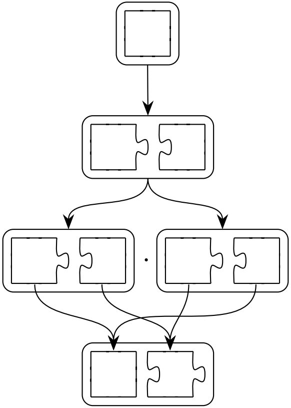
Perks of infinity: $e^{\omega} = (ee)^{\omega}$



instead of

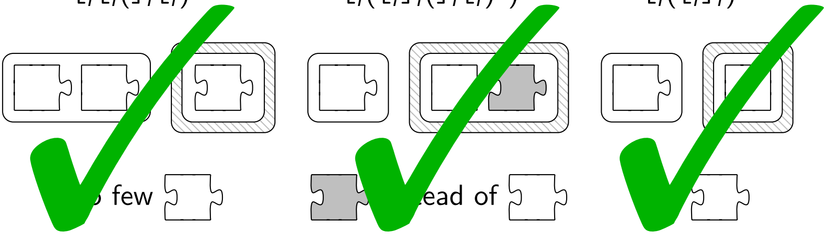


No



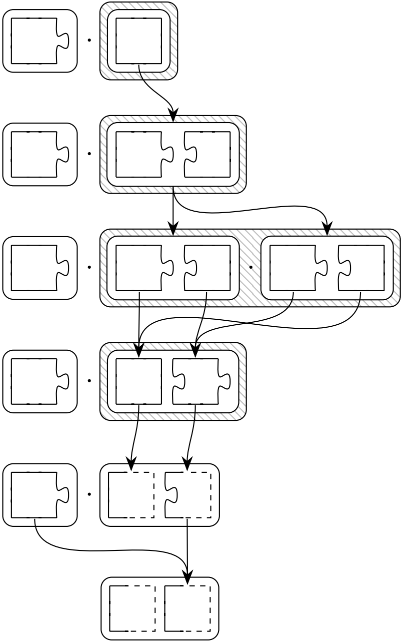
Problems

$[_i\, [_i (\,]_i\, [_i)^\omega$ $[_i ([_i]_i ( ]_i [_i)^*)^\omega$ $[_i ([_i]_i)^\omega$



Example

$[_i ([_i]_i)^\omega$



What's next?

- Context-free languages
- Binary shuffles

That's all folks!