Table of Contents

[Algorithms 2](#_Toc88678741)

[1. A linear time extension of deterministic pushdown automata 2](#_Toc88678742)

[2. Compiler algorithms 2](#_Toc88678743)

[3. Neural Turing Machine 2](#_Toc88678744)

[4. Neural state PDA 2](#_Toc88678745)

[5. Visible PDA 2](#_Toc88678746)

[6. Nondeterministic Stack RNNs 2](#_Toc88678747)

[7. The Neural Network Pushdown Automaton 2](#_Toc88678748)

[8. Context-Free Transductions with Neural Stacks 2](#_Toc88678749)

[9. Subtree Matching by Pushdown Automata⋆ 2](#_Toc88678750)

[10. Building a Turing Machine with Reinforcement Learning 2](#_Toc88678751)

[11. CYK\_algorithm extensions 2](#_Toc88678752)

[12. Probabilistic context-free grammar and inside-outside algorithm 2](#_Toc88678753)

[Ideas 3](#_Toc88678754)

[13. Turing Machines 3](#_Toc88678755)

[14. Compiler 3](#_Toc88678756)

[15. Natural language processing 3](#_Toc88678757)

Brainstormen:

# Algorithms

## A linear time extension of deterministic pushdown automata

<https://aclanthology.org/W09-4626.pdf>

## Compiler algorithms

<http://www.softpanorama.org/Algorithms/compilers.shtml>

## Neural Turing Machine

<https://en.wikipedia.org/wiki/Neural_Turing_machine>

<https://awesomeopensource.com/project/WhenDustSettles/Neural-Turing-Machine>

## Neural state PDA

<https://deepai.org/publication/the-neural-state-pushdown-automata>

## Visible PDA

<https://www.semanticscholar.org/paper/A-new-algorithm-for-the-determinisation-of-visibly-Pol%C3%A1ch-Travnicek/d6ec2c5eb7c71b98946c1525638529ced397ec5b>

## Nondeterministic Stack RNNs

<https://deepai.org/publication/learning-context-free-languages-with-nondeterministic-stack-rnns>

## The Neural Network Pushdown Automaton

<https://deepai.org/publication/the-neural-network-pushdown-automaton-model-stack-and-learning-simulations>

## Context-Free Transductions with Neural Stacks

<https://deepai.org/publication/context-free-transductions-with-neural-stacks>

## Subtree Matching by Pushdown Automata⋆

<http://elib.mi.sanu.ac.rs/files/journals/csis/14/070205.pdf>

## Building a Turing Machine with Reinforcement Learning

<https://towardsdatascience.com/building-a-turing-machine-with-reinforcement-learning-9d06a4f0ce6>

## CYK\_algorithm extensions

<https://en.wikipedia.org/wiki/CYK_algorithm#Extensions>

## Probabilistic context-free grammar and inside-outside algorithm

## Compiler algorithms

<http://www.softpanorama.org/Algorithms/compilers.shtml>

# Ideas

## Turing Machines

AI, GTA1, autopilot  
<https://ieeexplore.ieee.org/abstract/document/7727551>

Algorithms:

## Compiler

Gameboy

## Natural language processing

spellchecker

<https://www.analyticssteps.com/blogs/top-10-applications-natural-language-processing-nlp>

# FINAL CHOICE backup option

## VOICE ASSSISTANT calculator

Algorithms: speech to text, parser, calculator

# Final Choice 2.0 – Platinum

Racing game simulatie

1. Neural netwerk bestuurd de auto, traint op het afleggen van de track:  
   Uitbreidend algorithme: Neural network Turing machines, Reinforcement learning
2. Algoritme om afbeelding van een (getekende) track parsen naar een racetrack (in logica)
3. Eventeel extra uitbreidend algoritme om de specificaties van de auto te beschrijven.

# Mail

# **Beste meneer Hofkens,**

Voor ons project van machines en berekenbaarheid willen wij graag de volgende applicatie maken:

Wij willen een simpele racing game simuleren  en maken hier gebruik van de volgende algoritmes en functies

1. Neural netwerk bestuurd de auto, traint op het afleggen van de track:  
   Dit maakt gebruik van Neural network Turing machines en reinforcement learning . Wij waren aan het denken om dit met PyTorch te doen.
2. Algoritme om afbeelding van een (getekende) track parsen naar een racetrack (in logica)
3. Eventeel extra uitbreidend algoritme om de specificaties van de auto te beschrijven.