

INF2080 – Logikk og beregninger

Forelesning 6: Pushdown-automater



UiO : **Institutt for informatikk**

Sist oppdatert: 2012-01-31 17:14

6.1 Pushdown-automater

Grenser for DFA

Grenser for DFA

- Memory gitt på forhånd
- Uavhengig av input

Grenser for DFA

- Memory gitt på forhånd
- Uavhengig av input
- Gir rask beregning

Grenser for DFA

- Memory gitt på forhånd
- Uavhengig av input
- Gir rask beregning
- Finne memory — finne konfigurasjoner (intensjoner)

Grenser for DFA

- Memory gitt på forhånd
- Uavhengig av input
- Gir rask beregning
- Finne memory — finne konfigurasjoner (intensjoner)
- Problemer om memory er avhengig av input — tall, parenteser

Grenser for DFA

Grenser for DFA

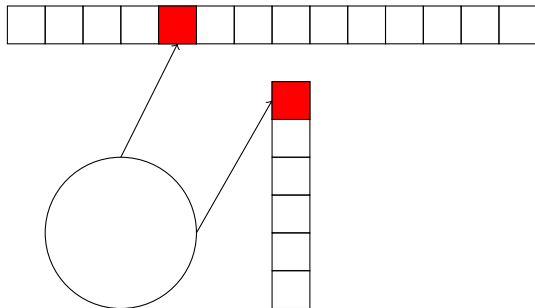
Push down automaton — stack automat

Horisontal input tape — leses fra venstre mot høyre

Vertikal stack

To aktive ruter — en på tape og den øverste på stack

Endelig kontroll



Definisjon

Definisjon

Input alfabet: Et endelig alfabet \mathcal{I}

Stack alfabet: Et endelig alfabet \mathcal{S}

Definisjon

Input alfabet: Et endelig alfabet \mathcal{I}

Stack alfabet: Et endelig alfabet \mathcal{S}

Tilstander: En endelig mengde tilstander \mathcal{Q}

Definisjon

Input alfabet: Et endelig alfabet \mathcal{I}

Stack alfabet: Et endelig alfabet \mathcal{S}

Tilstander: En endelig mengde tilstander \mathcal{Q}

Start: En start tilstand. Ved start er stacken tom.

Definisjon

Input alfabet: Et endelig alfabet \mathcal{I}

Stack alfabet: Et endelig alfabet \mathcal{S}

Tilstander: En endelig mengde tilstander \mathcal{Q}

Start: En start tilstand. Ved start er stacken tom.

Transisjoner: Mer om dette neste side. Transisjonene er ikke deterministiske.

Definisjon

Input alfabet: Et endelig alfabet \mathcal{I}

Stack alfabet: Et endelig alfabet \mathcal{S}

Tilstander: En endelig mengde tilstander \mathcal{Q}

Start: En start tilstand. Ved start er stacken tom.

Transisjoner: Mer om dette neste side. Transisjonene er ikke deterministiske.

Akseptering: En mengde finale tilstander $\mathcal{A} \subseteq \mathcal{Q}$. Et ord blir akseptert om det fins en kjøring som ender opp i en final tilstand med tom stack.

Transisjoner

Transisjonene er gitt ved en rekke piler mellom tilstandene. Hver pil inneholder fire informasjonsbiter

- Fra-tilstand

Transisjoner

Transisjonene er gitt ved en rekke piler mellom tilstandene. Hver pil inneholder fire informasjonsbiter

- Fra-tilstand
- Vokter

Transisjoner

Transisjonene er gitt ved en rekke piler mellom tilstandene. Hver pil inneholder fire informasjonsbiter

- Fra-tilstand
- Vokter
- Aksjon

Transisjoner

Transisjonene er gitt ved en rekke piler mellom tilstandene. Hver pil inneholder fire informasjonsbiter

- Fra-tilstand
- Vokter
- Aksjon
- Til-tilstand

Transisjoner

Transisjonene er gitt ved en rekke piler mellom tilstandene. Hver pil inneholder fire informasjonsbiter

- Fra-tilstand
- Vokter
- Aksjon
- Til-tilstand

På pila skriver vi **vokter;aksjon**. Vokter er betingelsen for å følge pila og aksjonene er resultatet av å følge den.

Transisjoner

Transisjonene er gitt ved en rekke piler mellom tilstandene. Hver pil inneholder fire informasjonsbiter

- Fra-tilstand
- Vokter
- Aksjon
- Til-tilstand

På pila skriver vi **vokter;aksjon**. Vokter er betingelsen for å følge pila og aksjonene er resultatet av å følge den.

Vokter: En vokter er et element i $(\mathcal{J} \cup \{\varepsilon\}) \times (\mathcal{S} \cup \{\varepsilon\})$. Vi må sjekke symbolene i de to aktive rutene — eventuelt ikke noe symbol om vi har ε .

Transisjoner

Transisjonene er gitt ved en rekke piler mellom tilstandene. Hver pil inneholder fire informasjonsbiter

- Fra-tilstand
- Vokter
- Aksjon
- Til-tilstand

På pila skriver vi **vokter;aksjon**. Vokter er betingelsen for å følge pila og aksjonene er resultatet av å følge den.

Vokter: En vokter er et element i $(\mathcal{J} \cup \{\varepsilon\}) \times (\mathcal{S} \cup \{\varepsilon\})$. Vi må sjekke symbolene i de to aktive rutene — eventuelt ikke noe symbol om vi har ε .

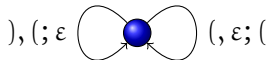
Aksjon: Et ord fra stack alfabetet \mathcal{S}^* som blir puttet på toppen av stacken. Den aktive ruta er toppen av stacken.

Parentessjekker

(() () (()

Stack:

Parentessjekker

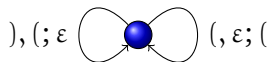


Let us see how the PDA checks an expression.

Input: () () (()

Stack: (

Parentessjekker

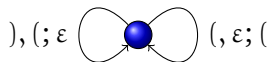


Let us see how the PDA checks an expression.

Input:) () (()

Stack: ((

Parentessjekker

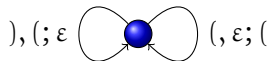


Let us see how the PDA checks an expression.

Input: () (()

Stack: (

Parentessjekker

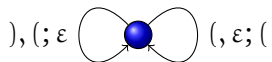


Let us see how the PDA checks an expression.

Input:) (()

Stack: ((

Parentessjekker

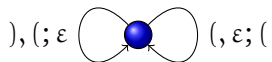


Let us see how the PDA checks an expression.

Input: (()

Stack: (

Parentessjekker

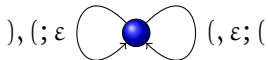


Let us see how the PDA checks an expression.

Input: ()

Stack: ((

Parentessjekker

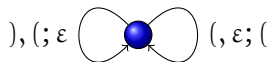


Let us see how the PDA checks an expression.

Input:)

Stack: (((

Parentessjekker



Let us see how the PDA checks an expression.

Input:

Stack: ((

Eksempler

Eksempler

- Parenteser
- Prosedyrekall

Eksempler

- Parenteser
- Prosedyrekall
- Kontekst frie språk

Eksempler

- Parenteser
- Prosedyrekall
- Kontekst frie språk
- Backus normal form

Eksempler

- Parenteser
- Prosedyrekall
- Kontekst frie språk
- Backus normal form
- HTML-kode

Eksempler

- Parenteser
- Prosedyrekall
- Kontekst frie språk
- Backus normal form
- HTML-kode
- LaTeX-kode

Eksempler

- Parenteser
- Prosedyrekall
- Kontekst frie språk
- Backus normal form
- HTML-kode
- LaTeX-kode
- ...

Eksempler

- Parenteser
- Prosedyrekall
- Kontekst frie språk
- Backus normal form
- HTML-kode
- LaTeX-kode
- ...

Rask så lenge vi kan gi plass til stacken på forhånd

Eksempler

- Parenteser
- Prosedyrekall
- Kontekst frie språk
- Backus normal form
- HTML-kode
- LaTeX-kode
- ...

Rask så lenge vi kan gi plass til stacken på forhånd
Begrenset antall typer parenteser