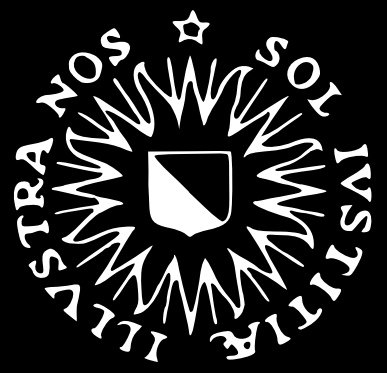
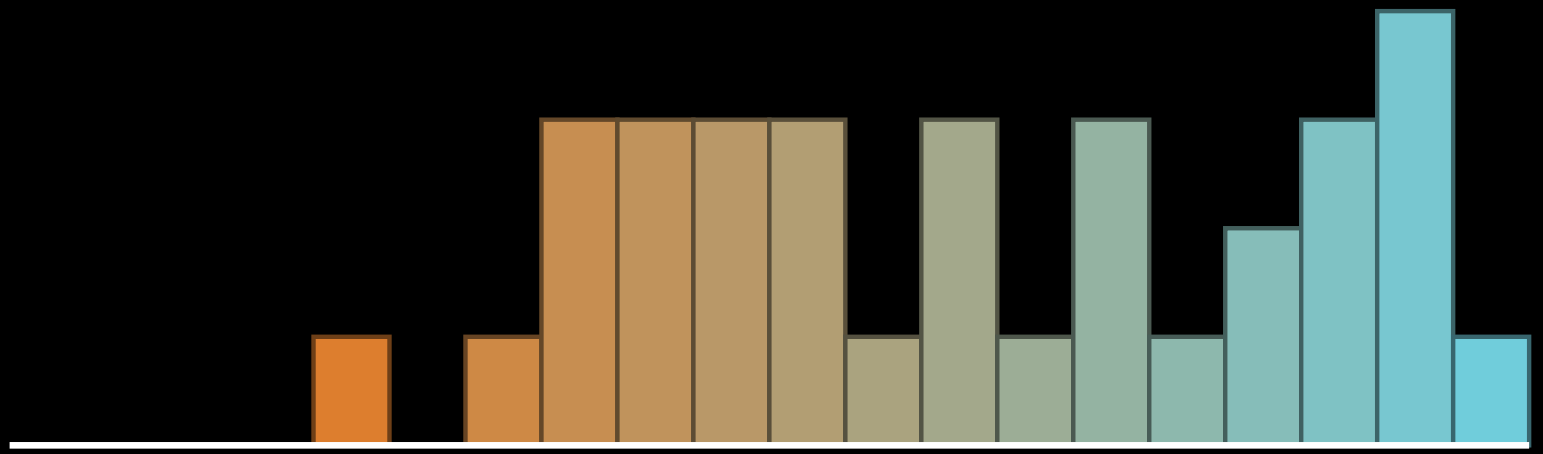


Logische Complexiteit

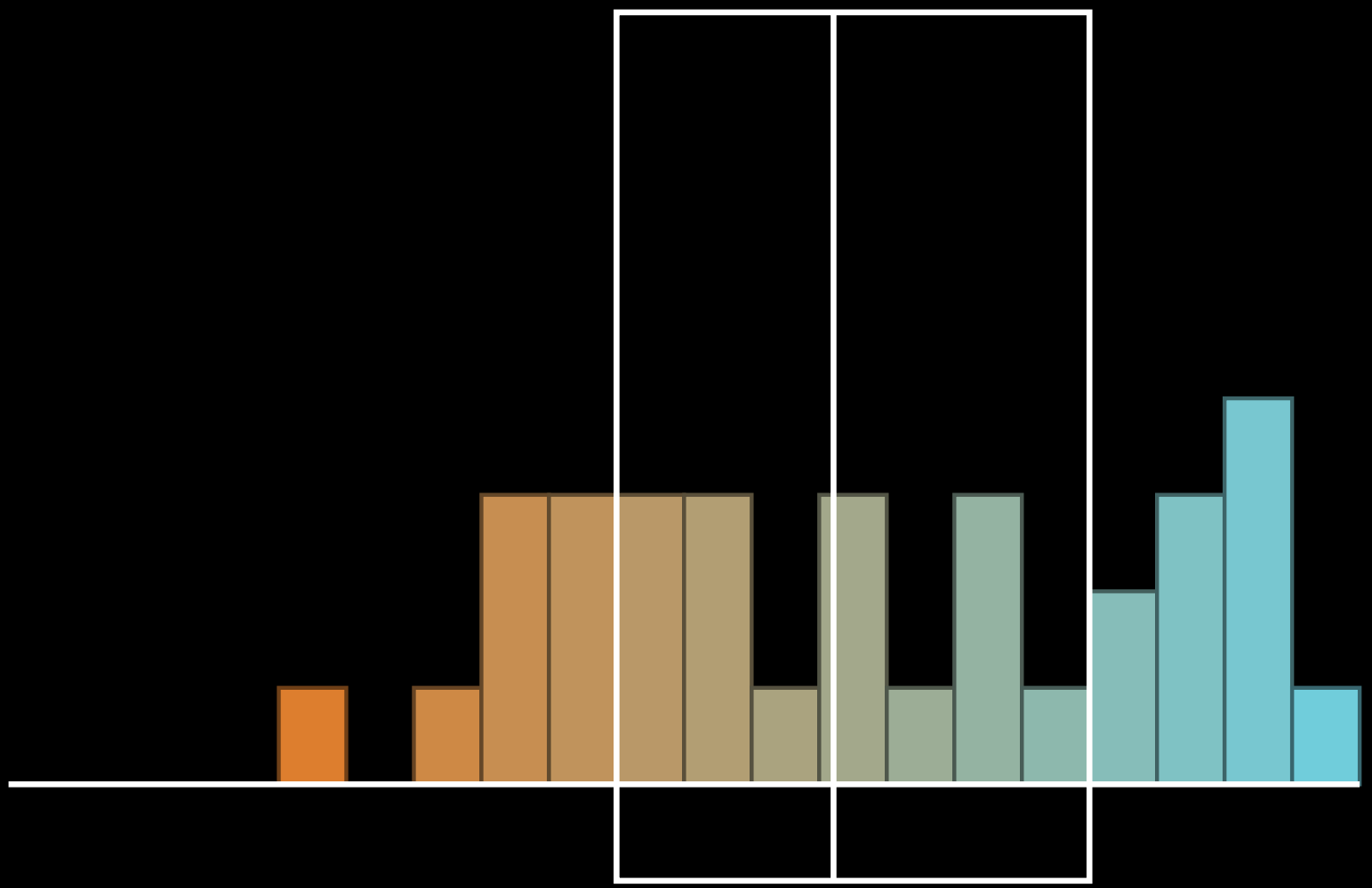
XIV : Stelling van Cook–Levin

Jeroen Goudsmit
Universiteit Utrecht
dinsdag 27 maart 2012

Inleveropgave 5

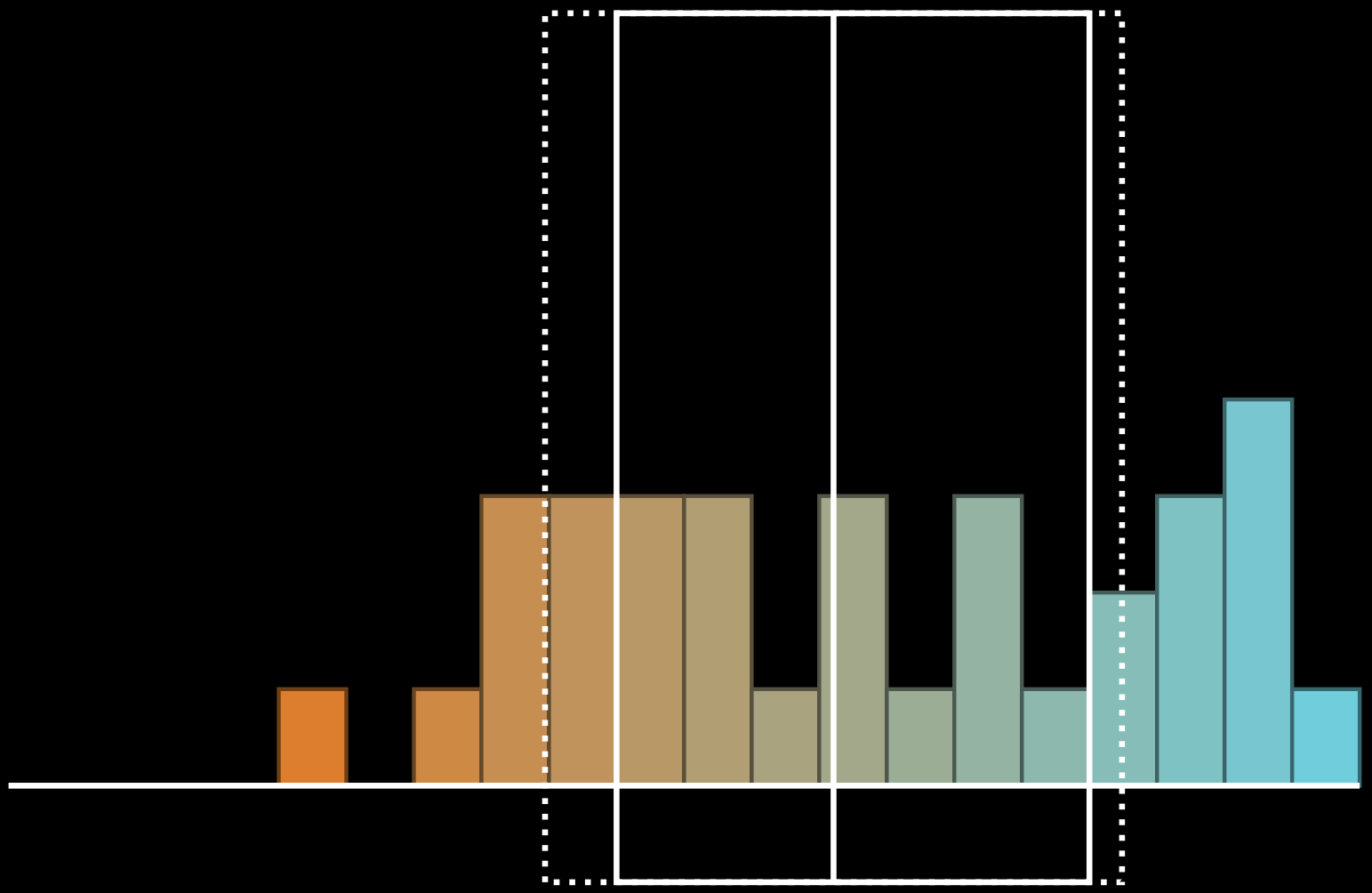


Inleveropgave 5



6.1

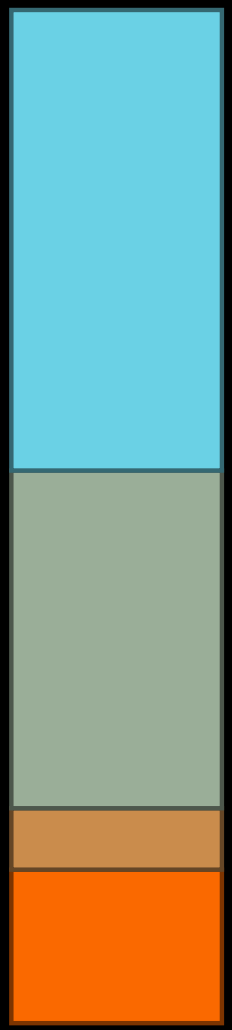
Inleveropgave 5



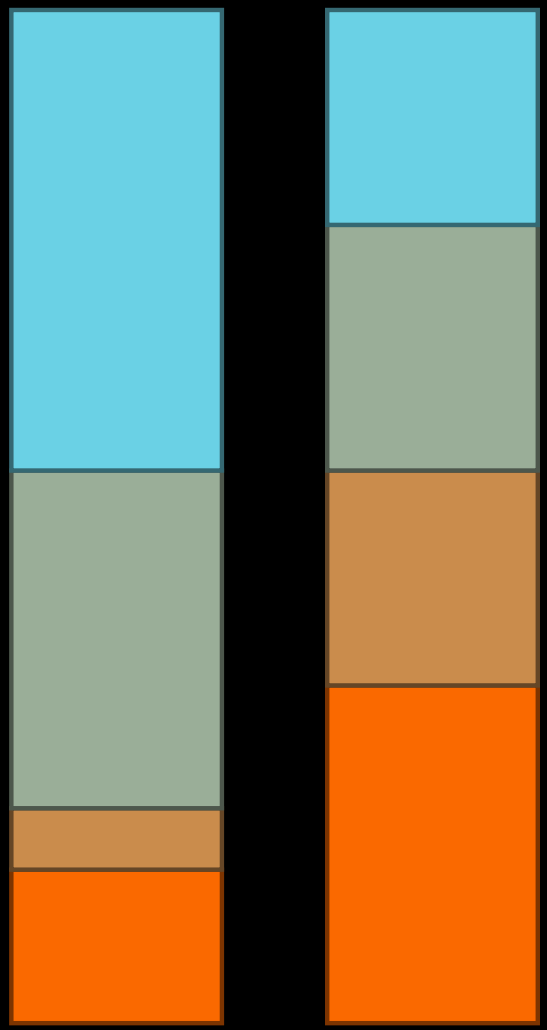
6.1

Inleveropgave 6

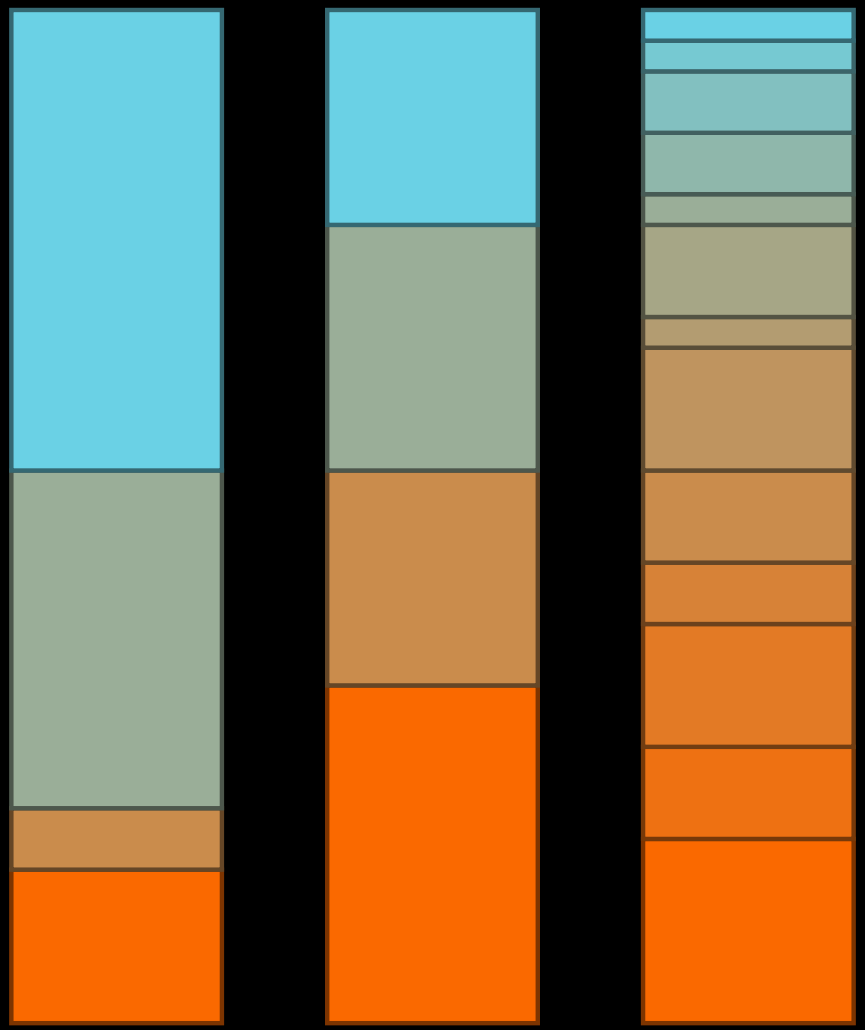
Inleveropgave 6



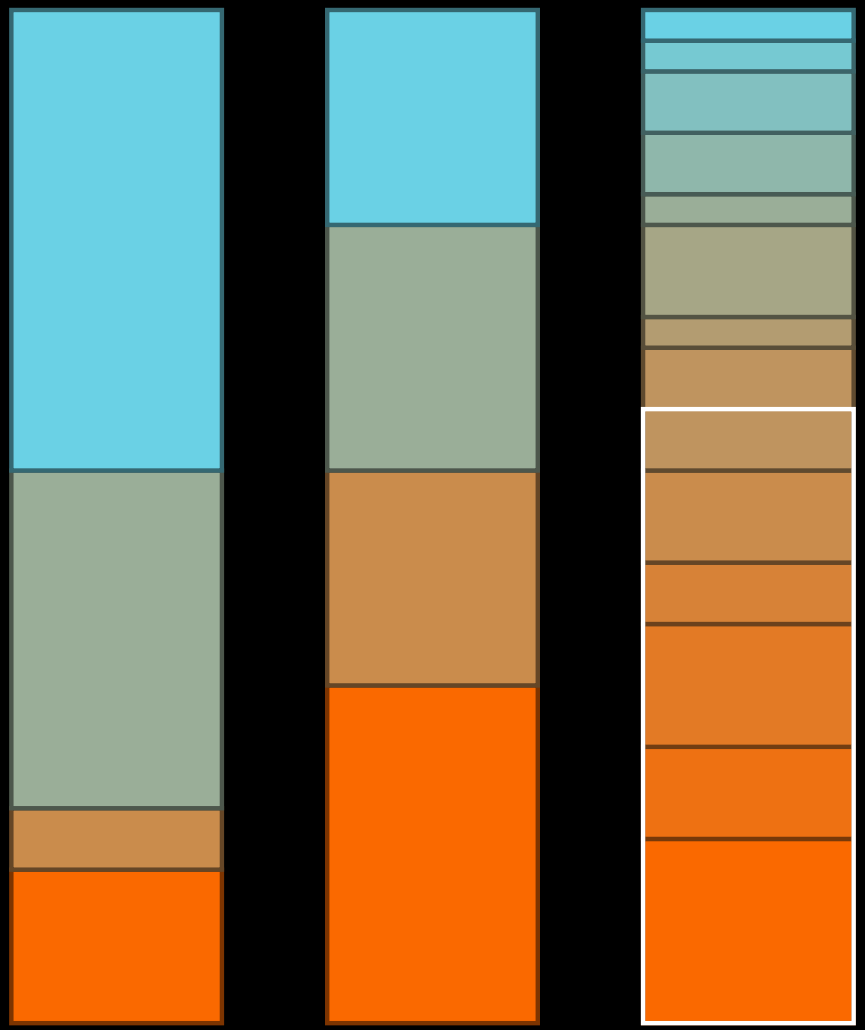
Inleveropgave 6



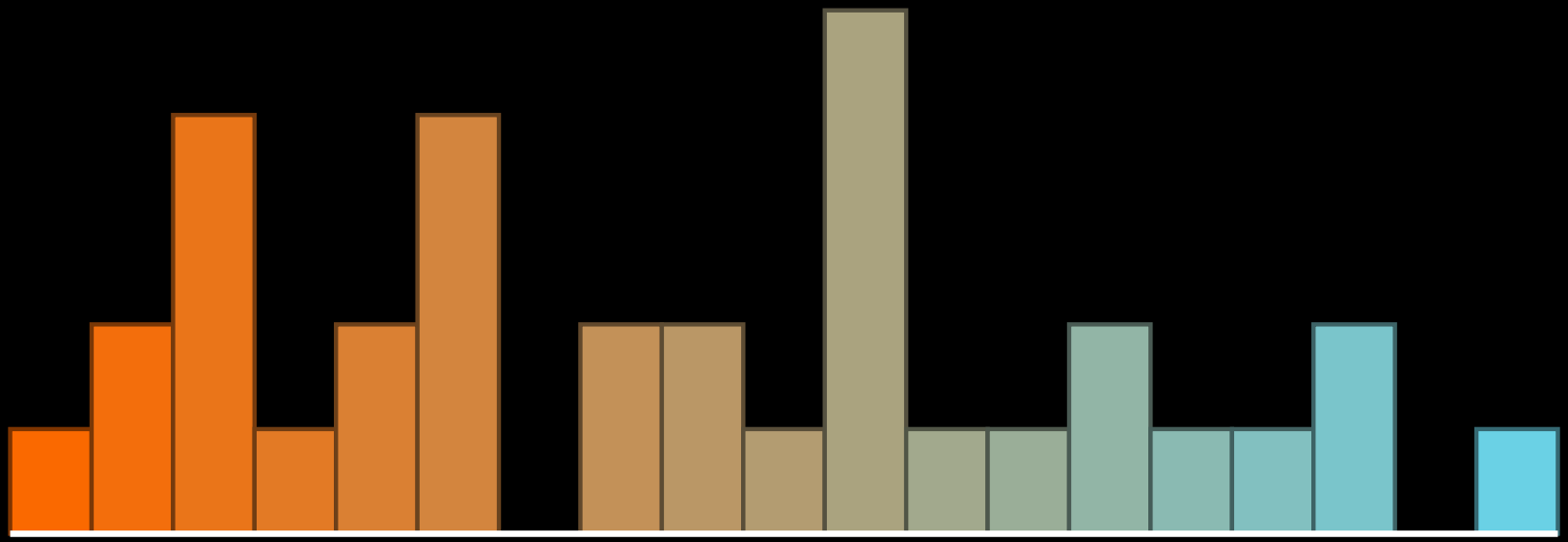
Inleveropgave 6



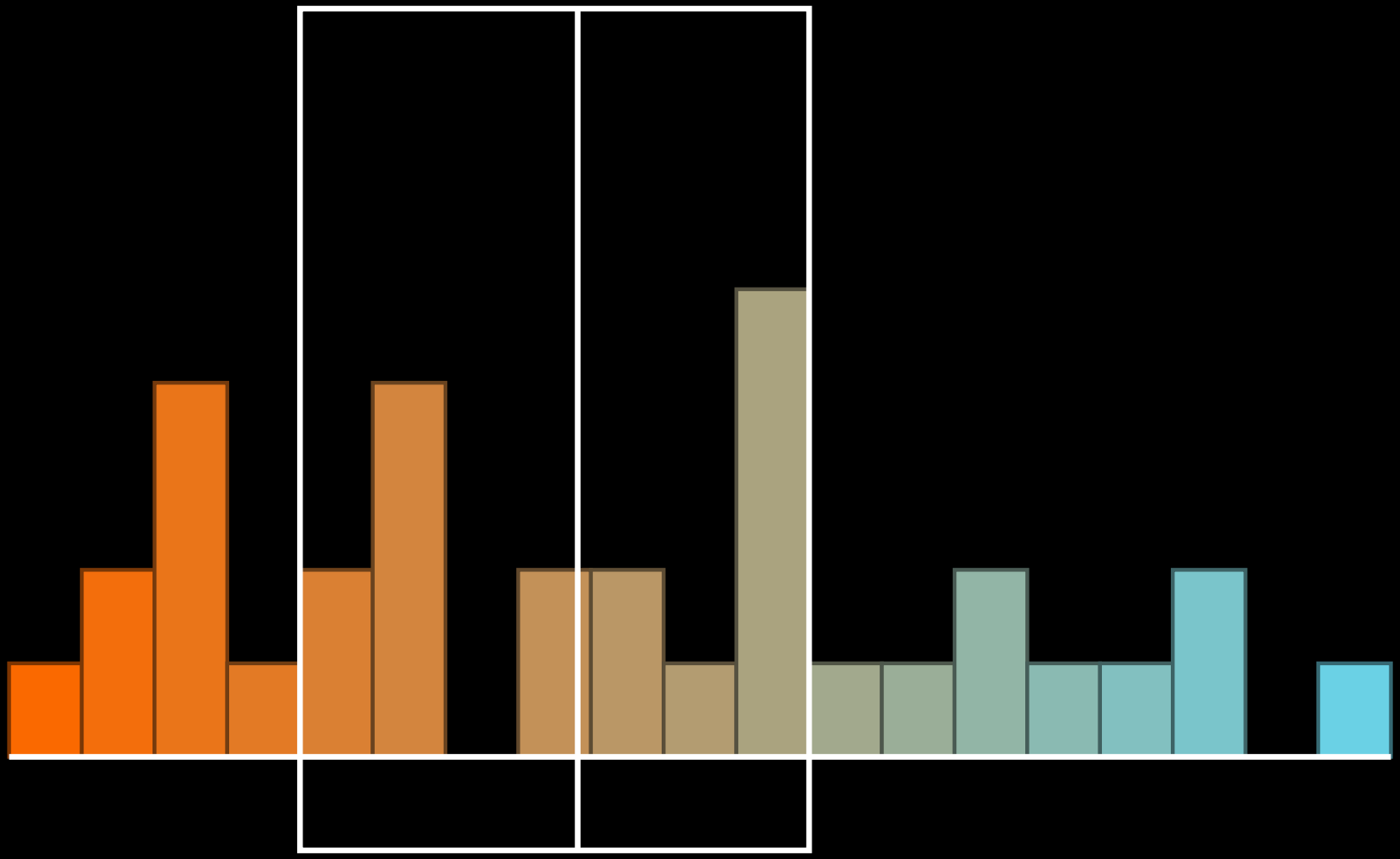
Inleveropgave 6



Inleveropgave 6

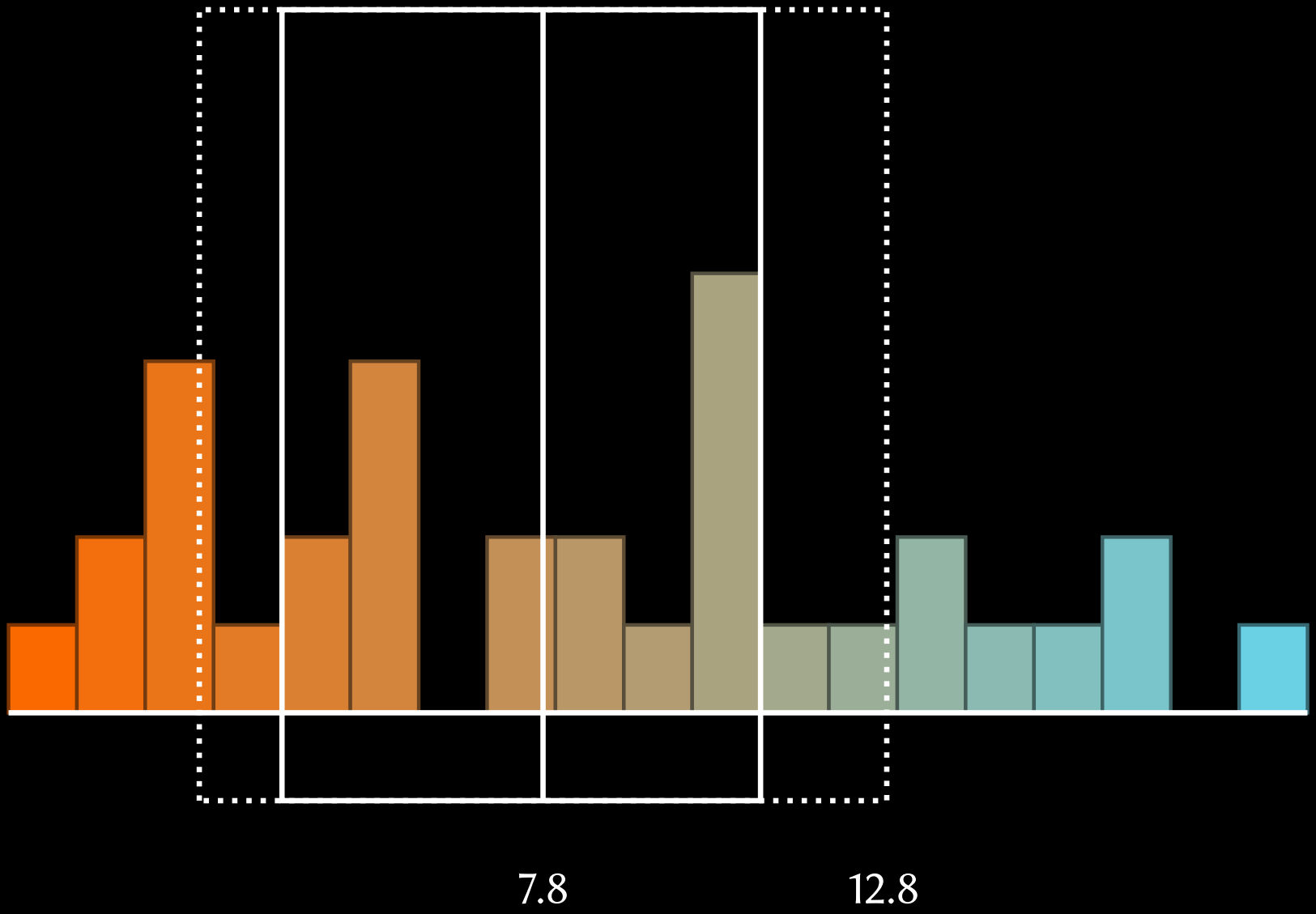


Inleveropgave 6



7.8

Inleveropgave 6



Cook–Levin Stelling

$$\text{Sat} \in \mathrm{P} \quad \text{precies als} \quad \mathrm{P} = \mathrm{NP}$$

Cook–Levin Stelling

Sat is NP-volledig

Cook–Levin Stelling

$$\forall A \in \mathbf{NP}.\ A \leq_{\mathrm{P}} \mathrm{Sat}$$

Bewijs

Gegeven nondeterministische poly-tijd TM

$$M = \left\langle Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}} \right\rangle$$

willen we een reductie van $\mathcal{L}(M)$ naar Sat.

Bewijs

Gegeven nondeterministische poly-tijd TM

$$M = \left\langle Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}} \right\rangle$$

willen we een reductie van $\mathcal{L}(M)$ naar Sat.

nondeterministische poly-tijd

M verwerkt w in $O(p(|w|))$ tijd.

Bewijs

Gegeven nondeterministische poly-tijd TM

$$M = \left\langle Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}} \right\rangle$$

willen we een reductie van $\mathcal{L}(M)$ naar Sat.

nondeterministische poly-tijd

M verwerkt w in $O(p(|w|))$ tijd:

$\exists N, c \in \mathbb{N}.\ \forall n \geq N.$ run-time van M op invoer van lengte $n \leq c \cdot p(n)$

Bewijs

Gegeven nondeterministische poly-tijd TM

$$M = \left\langle Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}} \right\rangle$$

willen we een reductie van $\mathcal{L}(M)$ naar Sat. Er is een polynoom $p(n)$ die de run-time van M begrensd.

Bewijs

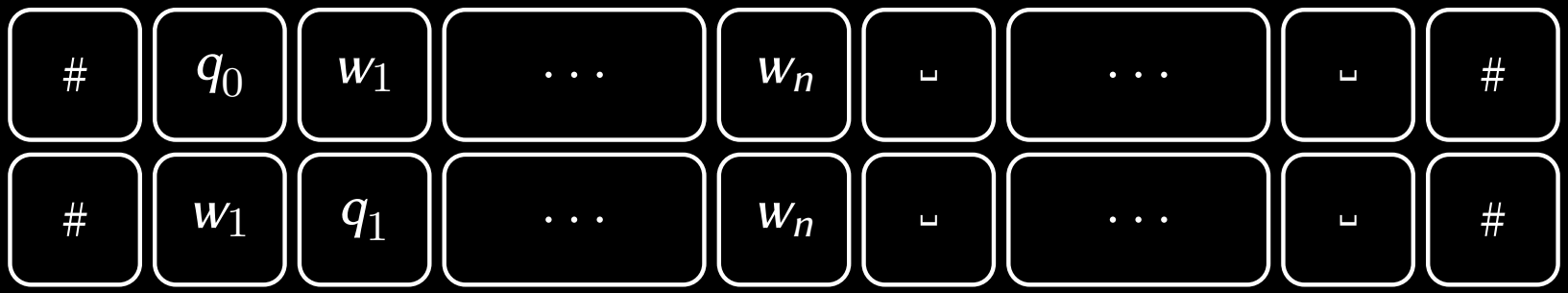
Gegeven nondeterministische poly-tijd TM

$$M = \left\langle Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}} \right\rangle$$

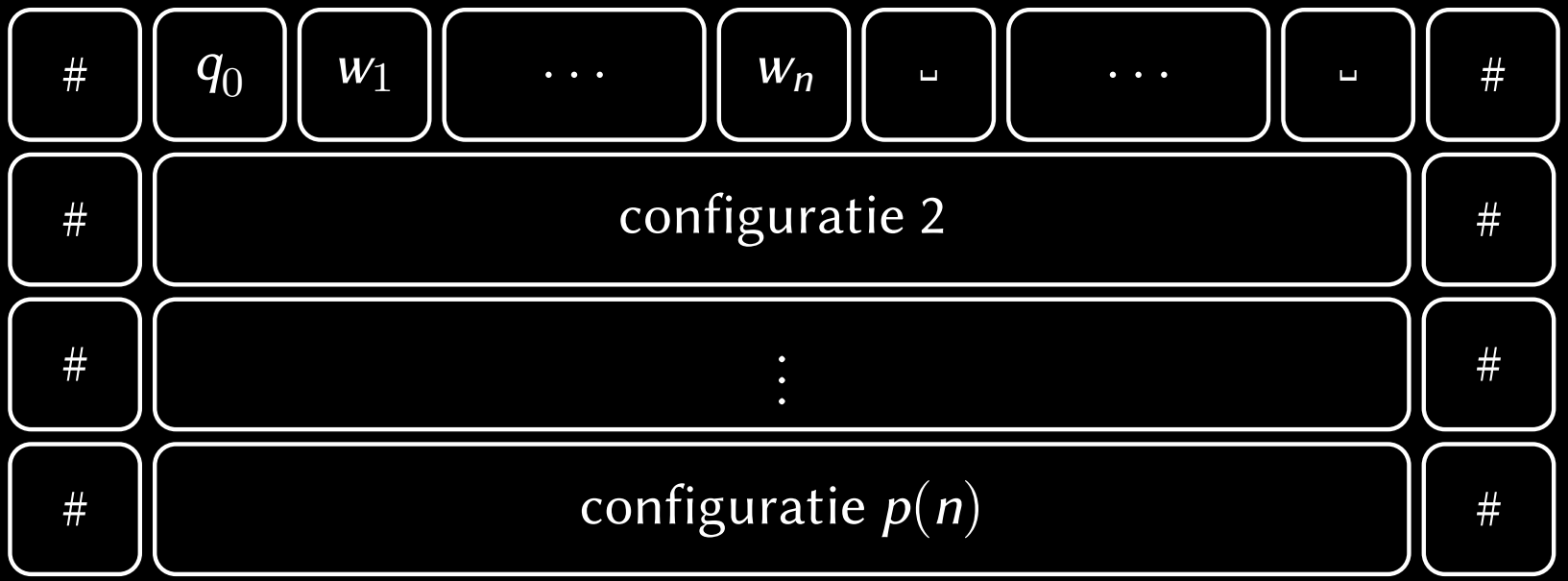
willen we een reductie van $\mathcal{L}(M)$ naar Sat. Er is een polynoom $p(n)$ die de run-time van M begrensd. Per $w \in \Sigma^*$, omschrijf hoe een accepterende berekening van M er uit ziet.

Configuraties

Configuraties



Configuraties



Codering

Neem $S := Q \cup \Gamma \cup \{\#\}$ en $K := \{0, \ldots, p(n)\}$.
Kijk naar een tableau van $p(n)$ bij $p(n)$, met waarden in S.

Variabelen $[i, j, s]$
$i, j \in K$ en $s \in S$.

Formule

1. enkelwaardigheid;
2. initiële configuratie;
3. acceptatie;
4. transitie.

Enkelwaardigheid

$$\phi_{\text{enkel}} := \bigwedge_{i,j \in K} \left(\left(\bigvee_{s \in S} [i, j, s] \right) \wedge \left(\bigwedge_{s \neq t \in S} \left(\neg [i, j, s] \vee \neg [i, j, t] \right) \right) \right)$$

Enkelwaardigheid

$$\phi_{\text{enkel}} := \bigwedge_{i,j \in K} \left(\left(\bigvee_{s \in S} [\, i, j, s \,] \right) \wedge \left(\bigwedge_{s \neq t \in S} \left(\neg [\, i, j, s \,] \vee \neg [\, i, j, t \,] \right) \right) \right)$$

Inhabitatie

Voor iedere $i, j \in K$ is er minstens één $s \in S$ met $[\, i, j, s \,]$ waar.

Enkelwaardigheid

Paarsgewijze Exclusiviteit

Per $i, j \in K$ is er hooguit één $s \in S$ met $[\,i, j, s\,]$ waar.

$$\phi_{\text{enkel}} := \bigwedge_{i,j \in K} \left(\left(\bigvee_{s \in S} [\,i, j, s\,] \right) \wedge \left(\bigwedge_{s \neq t \in S} \left(\neg [\,i, j, s\,] \vee \neg [\,i, j, t\,] \right) \right) \right)$$

Inhabitatie

Voor iedere $i, j \in K$ is er minstens één $s \in S$ met $[\,i, j, s\,]$ waar.

Initiële Configuratie

$$\phi_{\text{init}} := \begin{array}{l} [1,1,\#] \wedge [1,2,q_0] \wedge [1,3,w_1] \wedge \ldots \wedge [1,n+3,w_n] \\ \wedge\ [1,n+1,\text{␣}] \wedge \ldots \wedge [1,p(n)+3,\text{␣}] \wedge [1,p(n)+4,\#] \end{array}$$

Initiële Configuratie

markeer begin

$$\phi_{\text{init}} := [1,1,\#] \wedge [1,2,q_0] \wedge [1,3,w_1] \wedge \ldots \wedge [1,n+3,w_n]$$
$$\wedge\ [1,n+1,\textvisiblespace] \wedge \ldots \wedge [1,p(n)+3,\textvisiblespace] \wedge [1,p(n)+4,\#]$$

Initiële Configuratie

markeer begin

begintoestand

$$\phi_{\text{init}} := [1,1,\#] \wedge [1,2,q_0] \wedge [1,3,w_1] \wedge \ldots \wedge [1,n+3,w_n]$$

$$\wedge\ [1,n+1,\textvisiblespace] \wedge \ldots \wedge [1,p(n)+3,\textvisiblespace] \wedge [1,p(n)+4,\#]$$

Initiële Configuratie

markeer begin

begintoestand

invoer

$$[1,1,\#] \wedge [1,2,q_0] \wedge [1,3,w_1] \wedge \ldots \wedge [1,n+3,w_n]$$

$$\phi_{\text{init}} :=$$

$$\wedge\ [1,n+1,\textvisiblespace] \wedge \ldots \wedge [1,p(n)+3,\textvisiblespace] \wedge [1,p(n)+4,\#]$$

Initiële Configuratie

markeer begin

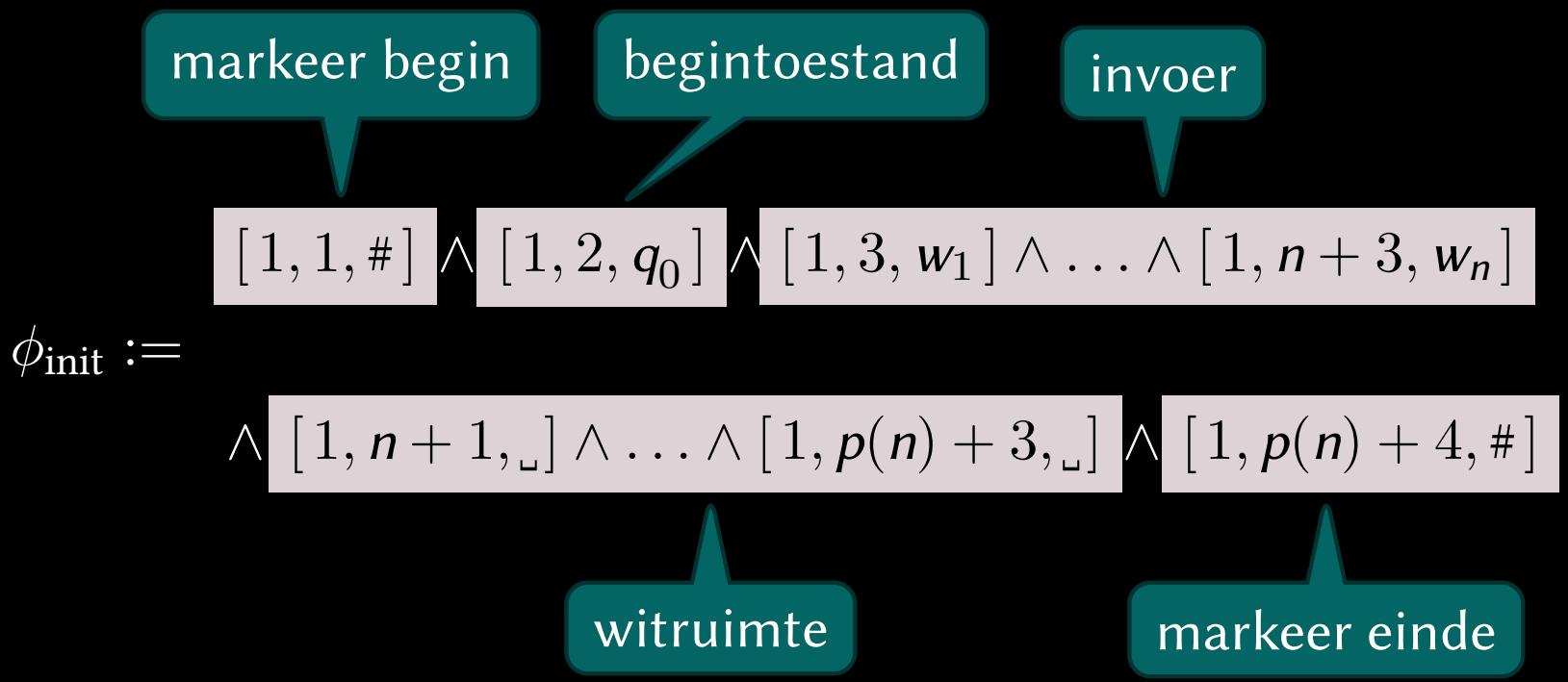
begintoestand

invoer

$$\phi_{\text{init}} := [1,1,\#] \wedge [1,2,q_0] \wedge [1,3,w_1] \wedge \ldots \wedge [1,n+3,w_n] \wedge [1,n+1,\textvisiblespace] \wedge \ldots \wedge [1,p(n)+3,\textvisiblespace] \wedge [1,p(n)+4,\#]$$

witruimte

Initiële Configuratie



Acceptatie

machine in eindtoestand

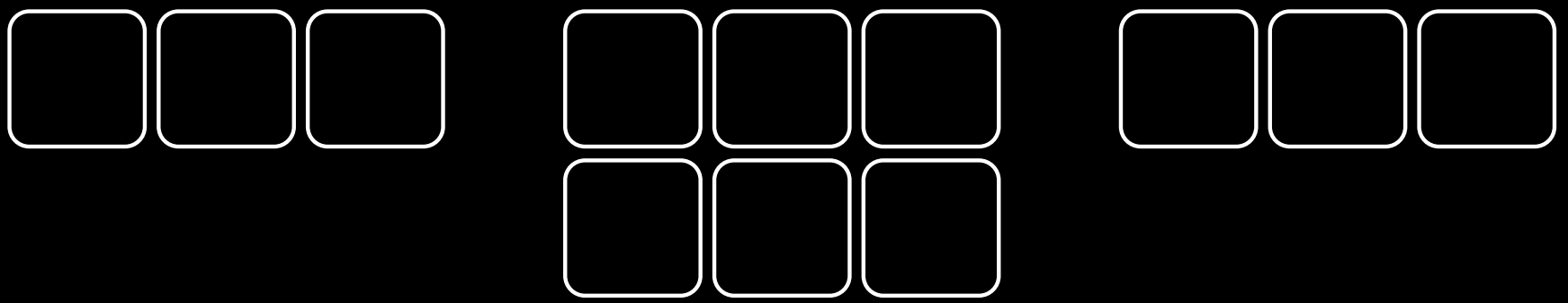
$$\phi_{\text{accept}} := \bigvee_{i,j \in K} \left[i, j, q_{\text{accept}} \right]$$

Acceptatie

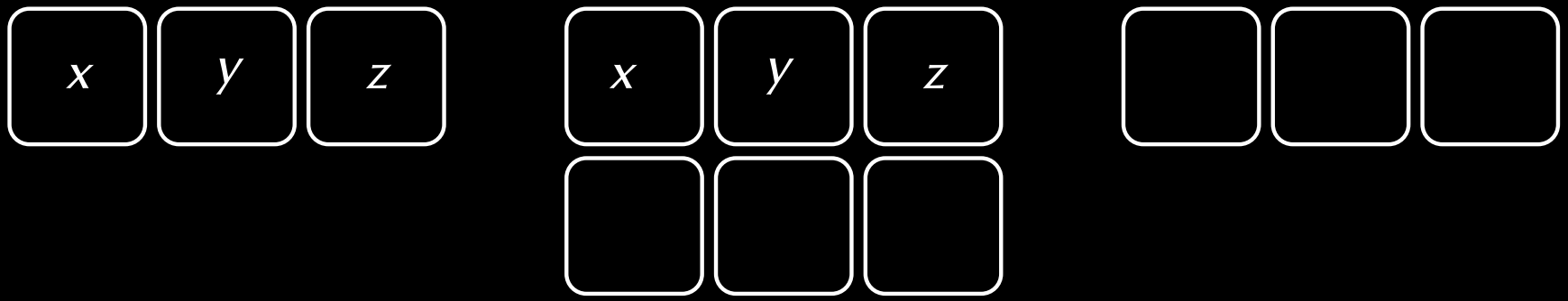
machine in eindtoestand

$$\phi_{\text{accept}} := \bigvee_{i,j \in K} \left[i, j, q_{\text{accept}} \right]$$

Transitie

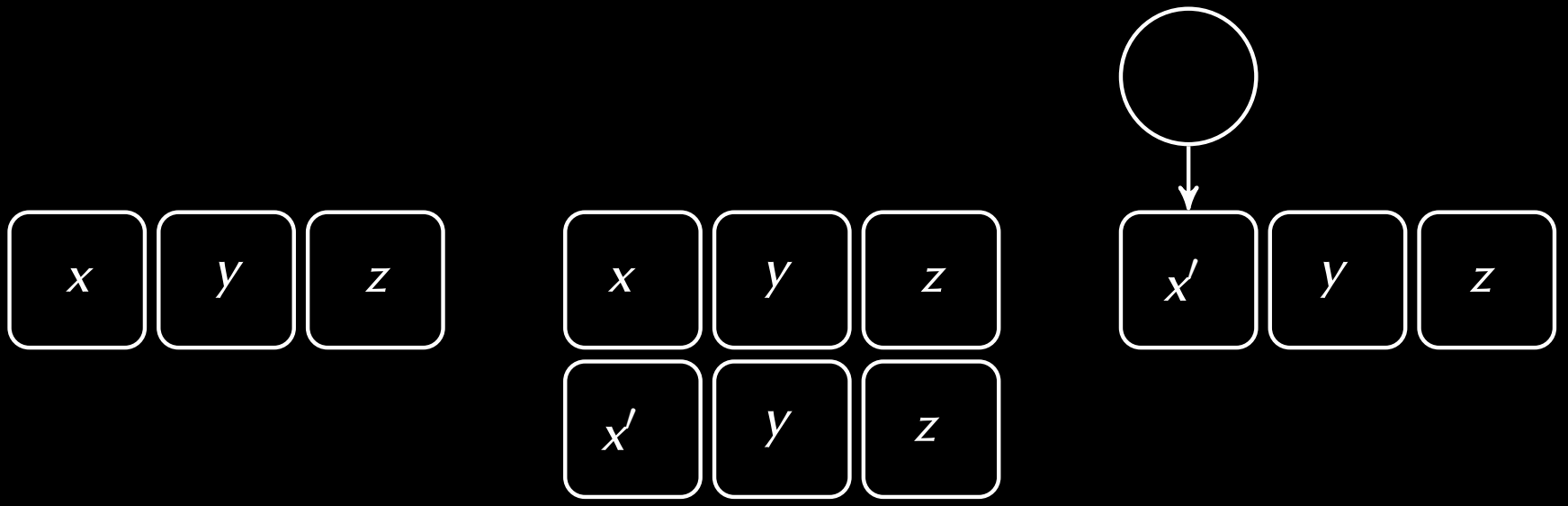


Transitie



$x, y, z \in \Gamma$

Transitie

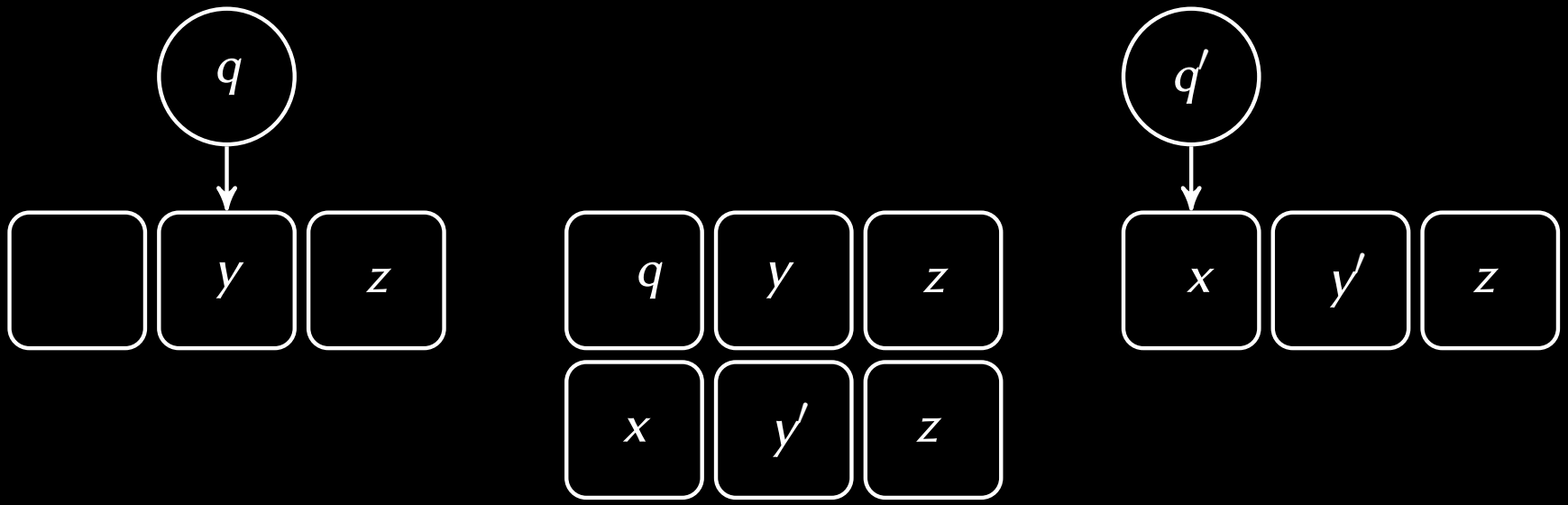


$x', x, y, z \in \Gamma$

Transitie

$q \in Q$ en $y, z \in \Gamma$

Transitie



$q', q \in Q$ en $x, y', y, z \in \Gamma$

Transitie

$q \in Q$ en y, $\quad z \in \Gamma$

Transitie

$q', q \in Q$ en $y, z', z \in \Gamma$

Transitie

$$\phi_{\text{trans}} := \bigwedge_{i,j \in K} \text{het venster van } i, j \text{ tot } i+2, j+1 \text{ klopt}$$

Transitie

correct venster

$$\phi_{\text{trans}} := \bigwedge_{i,j \in K} \text{het venster van } i, j \text{ tot } i+2, j+1 \text{ klopt}$$

Reductie

$$w \quad \mapsto \quad \left\langle \phi_{\text{enkel}} \wedge \phi_{\text{init}} \wedge \phi_{\text{accept}} \wedge \phi_{\text{trans}} \right\rangle$$

Reductie

$$w \quad \mapsto \quad \left\langle \phi_{\text{enkel}} \wedge \phi_{\text{init}} \wedge \phi_{\text{accept}} \wedge \phi_{\text{trans}} \right\rangle$$

Dit kan in $O\left(p(n)^2\right)$ tijd.

Linear Programmeren

$$\text{LinProg} := \left\{ \langle A, b, c, k \rangle \;\middle|\; \begin{array}{l} A, b \text{ en } c \text{ zijn matrices over } \mathbb{Z} \\ \text{zodat er een matrix } x \text{ over } \mathbb{Z} \text{ is} \\ \text{met } Ax \geq b \text{ en } x \cdot c \leq k \end{array} \right\}$$

Voorbeeld

$$\left\langle \begin{pmatrix} 0 & 4 & 0 \\ -3 & 2 & 1 \\ 0 & -1 & -9 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 4 \\ -20 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}, 3 \right\rangle \overset{?}{\in} \text{LinProg}$$

Voorbeeld

$$\left\langle \begin{pmatrix} 0 & 4 & 0 \\ -3 & 2 & 1 \\ 0 & -1 & -9 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 4 \\ -20 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}, 3 \right\rangle \overset{?}{\in} \text{LinProg}$$

$$\begin{pmatrix} 0 & 4 & 0 \\ -3 & 2 & 1 \\ 0 & -1 & -9 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \\ -19 \\ 0 \\ 1 \\ 2 \end{pmatrix} \geq \begin{pmatrix} 1 \\ 4 \\ -20 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Voorbeeld

$$\left\langle \begin{pmatrix} 0 & 4 & 0 \\ -3 & 2 & 1 \\ 0 & -1 & -9 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 4 \\ -20 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}, 3 \right\rangle \in \text{LinProg}$$

$$\begin{pmatrix} 0 & 4 & 0 \\ -3 & 2 & 1 \\ 0 & -1 & -9 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \\ -19 \\ 0 \\ 1 \\ 2 \end{pmatrix} \geq \begin{pmatrix} 1 \\ 4 \\ -20 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Ongelijkheden

$$
\begin{array}{rcrcrcl}
 & & 4x_1 & & & \geq & 1 \\
-3x_1 & + & 2x_2 & + & x_3 & \geq & 4 \\
 & & -x_2 & + & -9x_3 & \geq & -20 \\
x_1 & & & & & \geq & 0 \\
 & & x_2 & & & \geq & 0 \\
 & & & & x_3 & \geq & 0 \\
x_1 & + & x_2 & + & x_3 & \leq & 3
\end{array}
$$

NP-volledigheid van LinProg

3Sat reduceert tot LinProg in polynomiale tijd

Gegeven ϕ maken we vergelijkingen.

Gegeven ϕ maken we vergelijkingen. Zeg $a \in \psi$ als a voorkomt in ψ.

Gegeven ϕ maken we vergelijkingen. Zeg $a \in \psi$ als a voorkomt in ψ.
Per variabele $p \in \phi$, maak variabele x_p en $x_{\overline{p}}$ en ongelijkheden voor

$$0 \leq x_p, x_{\overline{p}} \leq 1 \quad \text{en} \quad x_{\overline{p}} + x_p = 1.$$

Gegeven ϕ maken we vergelijkingen. Zeg $a \in \psi$ als a voorkomt in ψ.
Per variabele $p \in \phi$, maak variabele x_p en $x_{\overline{p}}$ en ongelijkheden voor

$$0 \leq x_p, x_{\overline{p}} \leq 1 \quad \text{en} \quad x_{\overline{p}} + x_p = 1.$$

Per *clause* ψ in ϕ, maak de ongelijkheid

$$\sum_{a \in \psi} x_a \geq 1.$$

Gegeven ϕ maken we vergelijkingen. Zeg $a \in \psi$ als a voorkomt in ψ. Per variabele $p \in \phi$, maak variabele x_p en $x_{\overline{p}}$ en ongelijkheden voor

$$0 \leq x_p, x_{\overline{p}} \leq 1 \quad \text{en} \quad x_{\overline{p}} + x_p = 1.$$

Per *clause* ψ in ϕ, maak de ongelijkheid

$$\sum_{a \in \psi} x_a \geq 1.$$

Waren er n prop. variabelen en m *clauses*, nu zijn er $2n$ variabelen en $6n + m$ vergelijkingen.

Gegeven ϕ maken we vergelijkingen. Zeg $a \in \psi$ als a voorkomt in ψ.
Per variabele $p \in \phi$, maak variabele x_p en $x_{\overline{p}}$ en ongelijkheden voor

$$0 \leq x_p, x_{\overline{p}} \leq 1 \quad \text{en} \quad x_{\overline{p}} + x_p = 1.$$

Per *clause* ψ in ϕ, maak de ongelijkheid

$$\sum_{a \in \psi} x_a \geq 1.$$

Waren er n prop. variabelen en m *clauses*, nu zijn er $2n$ variabelen en $6n + m$ vergelijkingen.

Stel ϕ heeft een vervullende
bedeling, zeg v.

Gegeven ϕ maken we vergelijkingen. Zeg $a \in \psi$ als a voorkomt in ψ. Per variabele $p \in \phi$, maak variabele x_p en $x_{\overline{p}}$ en ongelijkheden voor

$$0 \leq x_p, x_{\overline{p}} \leq 1 \quad \text{en} \quad x_{\overline{p}} + x_p = 1.$$

Per *clause* ψ in ϕ, maak de ongelijkheid

$$\sum_{a \in \psi} x_a \geq 1.$$

Waren er n prop. variabelen en m *clauses*, nu zijn er $2n$ variabelen en $6n + m$ vergelijkingen.

Stel ϕ heeft een vervullende bedeling, zeg v. Per variabele $p \in \phi$, neem $x_p = 1$ precies als $v(p)$ waar is.

Gegeven ϕ maken we vergelijkingen. Zeg $a \in \psi$ als a voorkomt in ψ. Per variabele $p \in \phi$, maak variabele x_p en $x_{\overline{p}}$ en ongelijkheden voor

$$0 \leq x_p, x_{\overline{p}} \leq 1 \quad \text{en} \quad x_{\overline{p}} + x_p = 1.$$

Per *clause* ψ in ϕ, maak de ongelijkheid

$$\sum_{a \in \psi} x_a \geq 1.$$

Waren er n prop. variabelen en m *clauses*, nu zijn er $2n$ variabelen en $6n + m$ vergelijkingen.

Stel ϕ heeft een vervullende bedeling, zeg v. Per variabele $p \in \phi$, neem $x_p = 1$ precies als $v(p)$ waar is. Dit is een oplossing van de vergelijking.

Stel er is een oplossing x van de vergelijkingen.

Gegeven ϕ maken we vergelijkingen. Zeg $a \in \psi$ als a voorkomt in ψ. Per variabele $p \in \phi$, maak variabele x_p en $x_{\overline{p}}$ en ongelijkheden voor

$$0 \leq x_p, x_{\overline{p}} \leq 1 \quad \text{en} \quad x_{\overline{p}} + x_p = 1.$$

Per *clause* ψ in ϕ, maak de ongelijkheid

$$\sum_{a \in \psi} x_a \geq 1.$$

Waren er n prop. variabelen en m *clauses*, nu zijn er $2n$ variabelen en $6n + m$ vergelijkingen.

Stel ϕ heeft een vervullende bedeling, zeg v. Per variabele $p \in \phi$, neem $x_p = 1$ precies als $v(p)$ waar is. Dit is een oplossing van de vergelijking.

Stel er is een oplossing x van de vergelijkingen. Maak een bedeling v die $p \in \phi$ waar maakt exact als $x_p = 1$.

Gegeven ϕ maken we vergelijkingen. Zeg $a \in \psi$ als a voorkomt in ψ. Per variabele $p \in \phi$, maak variabele x_p en $x_{\overline{p}}$ en ongelijkheden voor

$$0 \leq x_p, x_{\overline{p}} \leq 1 \quad \text{en} \quad x_{\overline{p}} + x_p = 1.$$

Per *clause* ψ in ϕ, maak de ongelijkheid

$$\sum_{a \in \psi} x_a \geq 1.$$

Waren er n prop. variabelen en m *clauses*, nu zijn er $2n$ variabelen en $6n + m$ vergelijkingen.

Stel ϕ heeft een vervullende bedeling, zeg v. Per variabele $p \in \phi$, neem $x_p = 1$ precies als $v(p)$ waar is. Dit is een oplossing van de vergelijking.

Stel er is een oplossing x van de vergelijkingen. Maak een bedeling v die $p \in \phi$ waar maakt exact als $x_p = 1$. Elke *clause* wordt waargemaakt, dus ϕ is vervulbaar.

Ster-vrije Reguliere Expressies

$$
\begin{aligned}
\text{SterVrij} := \; & \emptyset \mid \varepsilon \mid \Sigma \\
& \mid \text{SterVrij}\ \text{SterVrij} \\
& \mid \text{SterVrij} \cup \text{SterVrij}
\end{aligned}
$$

Ster-vrije Inequivalentie

$$\text{SVIneq} := \left\{ \langle A, B \rangle \;\middle|\; \begin{array}{c} A \text{ en } B \text{ ster-vrije} \\ \text{inequivalente} \\ \text{reguliere expressies} \end{array} \right\}$$

Voorbeeld

$$\left\langle (1 \cup 1)0 \cup 01 \cup 10, (1 \cup 0) \cup 11 \right\rangle \stackrel{?}{\in} \mathbf{SVIneq}$$

Voorbeeld

$$\left\langle (1 \cup 1)0 \cup 01 \cup 10, (1 \cup 0) \cup 11 \right\rangle \in \mathbf{SVIneq}$$

Lemma
Er geldt SVIneq $\in$ NP.

Lemma
Er geldt SVIneq $\in$ NP.

Bewijs.

Beschouw de TM.
M = "Op invoer
$\langle \langle A, B \rangle, w \rangle$:

1. Test of $w \in \mathcal{L}(A)$.
2. Test of $w \in \mathcal{L}(B)$.
3. Als de antwoorden bij stap 1 en 2 verschillen, accepteer.

Lemma
Er geldt SVIneq $\in$ NP.

Bewijs.

Beschouw de TM.
M = "Op invoer
$\langle \langle A, B \rangle, w \rangle$:

1. Test of $w \in \mathcal{L}(A)$.
2. Test of $w \in \mathcal{L}(B)$.
3. Als de antwoorden bij stap 1 en 2 verschillen, accepteer.

Zie dat M in
polynomiale tijd werkt.

Lemma
Er geldt SVIneq $\in$ NP.

Bewijs.

Beschouw de TM.
M = "Op invoer $\langle \langle A, B \rangle, w \rangle$:

1. Test of $w \in \mathcal{L}(A)$.
2. Test of $w \in \mathcal{L}(B)$.
3. Als de antwoorden bij stap 1 en 2 verschillen, accepteer.

Zie dat M in polynomiale tijd werkt.

Als $\langle A, B \rangle \in$ SVIneq, dan is er een w met $|w| \leq |A|, |B|$ wat niet door beiden geaccepteerd wordt. Dus is w een getuige, zodat $\langle \langle A, B \rangle, w \rangle$ geaccepteerd wordt door M.

Lemma
Er geldt SVIneq $\in$ NP.

Bewijs.

Beschouw de TM.
M = "Op invoer $\langle \langle A, B \rangle, w \rangle$:

1. Test of $w \in \mathcal{L}(A)$.
2. Test of $w \in \mathcal{L}(B)$.
3. Als de antwoorden bij stap 1 en 2 verschillen, accepteer.

Zie dat M in polynomiale tijd werkt.

Als $\langle A, B \rangle \in$ SVIneq, dan is er een w met $|w| \leq |A|, |B|$ wat niet door beiden geaccepteerd wordt. Dus is w een getuige, zodat $\langle \langle A, B \rangle, w \rangle$ geaccepteerd wordt door M.

Als $\langle A, B \rangle \notin$ SVIneq, dan accepteren A en B dezelfde woorden. Dus is er geen w zodat 1 en 2 uit M verschillende antwoorden geven.

Lemma
Er geldt SVIneq $\in$ NP.

Bewijs.

Beschouw de TM.
M = "Op invoer $\langle \langle A, B \rangle, w \rangle$:

1. Test of $w \in \mathcal{L}(A)$.
2. Test of $w \in \mathcal{L}(B)$.
3. Als de antwoorden bij stap 1 en 2 verschillen, accepteer.

Zie dat M in polynomiale tijd werkt.

Als $\langle A, B \rangle \in$ SVIneq, dan is er een w met $|w| \leq |A|, |B|$ wat niet door beiden geaccepteerd wordt. Dus is w een getuige, zodat $\langle \langle A, B \rangle, w \rangle$ geaccepteerd wordt door M.

Als $\langle A, B \rangle \notin$ SVIneq, dan accepteren A en B dezelfde woorden. Dus is er geen w zodat 1 en 2 uit M verschillende antwoorden geven.

Lemma
Er geldt SVIneq $\in$ NP.

Bewijs.

Beschouw de TM.
M = "Op invoer $\langle \langle A, B \rangle, w \rangle$:
1. Test of $w \in \mathcal{L}(A)$.
2. Test of $w \in \mathcal{L}(B)$.
3. Als de antwoorden bij stap 1 en 2 verschillen, accepteer.

Zie dat M in polynomiale tijd werkt.

Als $\langle A, B \rangle \in$ SVIneq, dan is er een w met $|w| \leq |A|, |B|$ wat niet door beiden geaccepteerd wordt. Dus is w een getuige, zodat $\langle \langle A, B \rangle, w \rangle$ geaccepteerd wordt door M.

Als $\langle A, B \rangle \notin$ SVIneq, dan accepteren A en B dezelfde woorden. Dus is er geen w zodat 1 en 2 uit M verschillende antwoorden geven.

Dus M is een poly-tijd controleur van SVIneq. ■

Bedelingen

Bekijk een formule $\phi = \phi_1 \wedge \ldots \phi_n$ met variabelen $p_1, \ldots, p_m$.

Bedelingen

Bekijk een formule $\phi = \phi_1 \wedge \ldots \phi_n$ met variabelen $p_1, \ldots, p_m$. Per *clause* ϕ_i maken we een reguliere expressie als volgt.

$$F_i := \prod_{j=1}^{m} \begin{cases} 0 & \text{als } p_j \text{ in } \phi_i \\ 1 & \text{als } \neg p_j \text{ in } \phi_i \\ (1+0) & \text{anders} \end{cases}$$

Bedelingen

Bekijk een formule $\phi = \phi_1 \wedge \ldots \phi_n$ met variabelen $p_1, \ldots, p_m$. Per *clause* ϕ_i maken we een reguliere expressie als volgt.

$$F_i := \prod_{j=1}^{m} \begin{cases} 0 & \text{als } p_j \text{ in } \phi_i \\ 1 & \text{als } \neg p_j \text{ in } \phi_i \\ (1+0) & \text{anders} \end{cases}$$

Merk op dat $w_1 \ldots w_m \in \mathcal{L}(F_i)$ precies als $v(p_j) = w_j$ de *clause* ϕ_i niet waar maakt.

Bedelingen

Bekijk een formule $\phi = \phi_1 \wedge \ldots \phi_n$ met variabelen $p_1, \ldots, p_m$. Per *clause* ϕ_i maken we een reguliere expressie als volgt.

$$F_i := \prod_{j=1}^{m} \begin{cases} 0 & \text{als } p_j \text{ in } \phi_i \\ 1 & \text{als } \neg p_j \text{ in } \phi_i \\ (1+0) & \text{anders} \end{cases}$$

Merk op dat $w_1 \ldots w_m \in \mathcal{L}\left(F_i\right)$ precies als $v(p_j) = w_j$ de *clause* ϕ_i niet waar maakt. Een bedeling maakt ϕ niet waar precies als ze komt uit

$$A := \bigcup_{i=1}^{n} F_i.$$

Bedelingen

Bekijk een formule $\phi = \phi_1 \wedge \ldots \phi_n$ met variabelen $p_1, \ldots, p_m$. Per *clause* ϕ_i maken we een reguliere expressie als volgt.

$$F_i := \prod_{j=1}^{m} \begin{cases} 0 & \text{als } p_j \text{ in } \phi_i \\ 1 & \text{als } \neg p_j \text{ in } \phi_i \\ (1+0) & \text{anders} \end{cases}$$

Merk op dat $w_1 \ldots w_m \in \mathcal{L}(F_i)$ precies als $v(p_j) = w_j$ de *clause* ϕ_i niet waar maakt. Een bedeling maakt ϕ niet waar precies als ze komt uit

$$A := \bigcup_{i=1}^{n} F_i.$$

Zie dat A slechts hooguit $O(n \cdot m)$ groot is.

SVIneq is NP-volledig

Lemma
3Sat *reduceert tot* SVIneq *in polynomiale tijd.*

SVIneq is NP-volledig

Lemma
3Sat *reduceert tot* SVIneq *in polynomiale tijd.*

Bewijs.

Gegeven een formule ϕ, maak A voor alle *falende bedelingen* en B de reguliere expressie die alle bedelingen accepteert.

SVIneq is NP-volledig

Lemma

3Sat *reduceert tot* SVIneq *in polynomiale tijd.*

Bewijs.

Gegeven een formule ϕ, maak A voor alle *falende bedelingen* en B de reguliere expressie die alle bedelingen accepteert. Nu zit $\langle A, B \rangle$ in SVIneq enkel en alleen als ϕ vervulbaar is. ■

Gezien

NP-volledige problemen

Sat 3Sat Clique LinProg SVIneq