

Luento 6: Laskennallisesta vaativuudesta

1. Looginen päättely resoluutiolla (luennolta 5)
2. Laskennan malli
3. Keskeiset vaativuusluokat
4. Redusoituvuus ja vaikeat ongelmat

Loogiset päätelmät resoluutiolla

Muut loogiset päätelmät voidaan suorittaa seuraavasti:

1. Lause ϕ on *pätevä* $\iff$ $\text{KM}(\neg\phi)$ on kumoutuva.
2. Lause ϕ on *looginen seuraus* lausejoukosta Σ $\iff$ $\text{KM}(\Sigma \cup \{\neg\phi\})$ on kumoutuva.
3. Lauseet ϕ ja ψ ovat *loogisesti ekvivalentit* $\iff$ $\text{KM}(\{\phi, \neg\psi\})$ ja $\text{KM}(\{\neg\phi, \psi\})$ ovat kumpikin kumoutuvia.

Sen sijaan (vasta)mallien hakeminen resoluutiolla on hankalaa!

1. LOOGINEN PÄÄTTELY RESOLUUTIOLLA

Päättelyn lähtökohdaksi tarvitaan sopivia klausuulimuotoja:

- Lause $\phi \in \mathcal{L}$ muunnetaan konjunkttiiviseen normaalimuotoon ja edelleen klausuulijoukoksi, jota merkitään $\text{KM}(\phi)$.
- Lausejoukko $\Sigma \subseteq \mathcal{L}$ muunnetaan klausuulijoukoksi

$$\text{KM}(\Sigma) = \bigcup_{\phi \in \Sigma} \text{KM}(\phi).$$

Näitä voidaan käyttää suoraan toteutuvuuden tutkimiseen:

1. Lause ϕ on *toteutumaton* $\iff$ $\text{KM}(\phi)$ on kumoutuva.
2. Lausejoukko Σ on *toteutumaton* $\iff$ $\text{KM}(\Sigma)$ on kumoutuva.

Pätevyyden tutkiminen

Esimerkki 7.26 Tarkastellaan lausetta $(A \rightarrow B) \wedge (\neg A \rightarrow C) \rightarrow B \vee C$.

Lauseen *negation* KNM on $(\neg A \vee B) \wedge (A \vee C) \wedge \neg B \wedge \neg C$, joten vastaava klausuulijoukko on $S = \{\{\neg A, B\}, \{A, C\}, \{\neg B\}, \{\neg C\}\}$.

Resoluutiotodistus:

1.	$\{\neg A, B\}$	S
2.	$\{A, C\}$	S
3.	$\{\neg B\}$	S
4.	$\{\neg C\}$	S
5.	$\{B, C\}$	1, 2
6.	$\{C\}$	3, 5
7.	$\square$	4, 6

$\implies$ Lause on pätevä. ■

Pätevyyden tutkiminen

Esimerkki 7.27 Tutkitaan lausetta $(A \rightarrow B) \wedge (A \rightarrow C) \rightarrow B \vee C$.

Lauseen *negation* KNM on $(\neg A \vee B) \wedge (\neg A \vee C) \wedge \neg B \wedge \neg C$.

Resoluutiotodistus:

1. $\{\neg A, B\}$ S
2. $\{\neg A, C\}$ S
3. $\{\neg B\}$ S
4. $\{\neg C\}$ S
5. $\{\neg A\}$ 1, 3

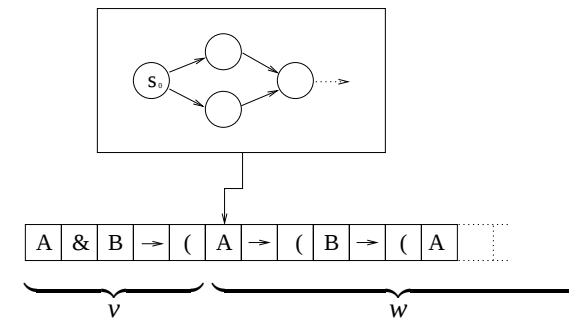
Muita klausuuleja (mukaan lukien $\square$) ei ole enää johdettavissa.

$\Rightarrow$ Lause ei ole pätevä.

Vastamalli $\mathcal{A} = \emptyset$ nähtävissä klausuuleista 3–5, kun $\mathcal{P} = \{A, B, C\}$ ■

2. LASKENNAN MALLI

Määritelmä 8.1 Deterministinen Turing-kone T on nelikkö $\langle A, S, s_0, t \rangle$, missä A on *aakkosto*, S on *tilajoukko*, $s_0 \in S$ on *alkutila* ja $t : S \times A \rightarrow S \times A \times \{\rightarrow, \leftarrow, \downarrow\}$ on *tilansiirtofunktio*.



Loogisen seuraavuuden tutkiminen

Esimerkki 7.30 Hissiesimerkin lauseille ja turvallisuusominaisuuden $\neg(A_1 \wedge A_2)$ *negatiolle* saadaan kokonaisuutena klausuulijoukko

$$S = \{\{\neg K_1, \neg K_2\}, \{\neg A_1, K_1\}, \{\neg A_2, K_2\}, \{A_1\}, \{A_2\}\}.$$

Joukon S hylkäyksestä muodostuu seuraava:

1. $\{\neg K_1, \neg K_2\}$	S	6. $\{K_1\}$	2, 4
2. $\{\neg A_1, K_1\}$	S	7. $\{K_2\}$	3, 5
3. $\{\neg A_2, K_2\}$	S	8. $\{\neg K_2\}$	1, 6
4. $\{A_1\}$	S	9. $\square$	7, 8
5. $\{A_2\}$	S		

$\Rightarrow$ Lause $\neg(A_1 \wedge A_2)$ on muiden lauseiden looginen seuraus. ■

Konfiguraatit ja laskennat

- Turing-koneen T kokonaistilan määrää *konfiguraatio* $\langle s, v, w \rangle$, missä $s \in S$ on koneen T tila sekä $v \in A^*$ ja $w \in A^+$ ovat merkkijonoja.
- Työnauhan sisältö on kokonaisuudessaan vw ja T käsittelee aina merkkijonon w ensimmäistä merkkiä (luku-kirjoituspuheen kohta).
- Tilasiirtymä $\langle s, v, w \rangle \xrightarrow{T} \langle s', v', w' \rangle$ riippuu tilansiirtofunktion arvosta $t(s, a) = \langle s', a', m \rangle$, missä a on w :n ensimmäinen merkki.

Määritelmä 8.2 Turing-koneen T suorittama n askeleen mittainen *laskenta* syötteellä w_0 on konfiguraatioiden jono

$$\langle s_0, v_0, w_0 \rangle \xrightarrow{T} \langle s_1, v_1, w_1 \rangle \xrightarrow{T} \dots \xrightarrow{T} \langle s_n, v_n, w_n \rangle,$$

missä $v_0 = \varepsilon$ ja s_n on yksi tilajoukon S *erikoistiloista* k , e tai p .

Hyväksyvä laskenta

- Deterministinen Turing-kone T **hyväksyy** syötteensä w_0 , joss $s_n = k$ koneen T suorittamalle laskennalle $\langle s_0, \varepsilon, w_0 \rangle \xrightarrow{T} \dots \xrightarrow{T} \langle s_n, v_n, w_n \rangle$.

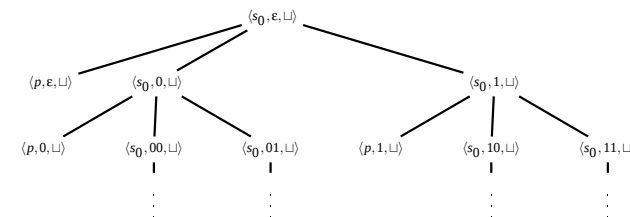
Esimerkki 8.3 Olkoon $A = \{0, 1, \sqcup\}$ ja $S = \{s_0, s_1, k, e, p\}$. Binääriluvun pariteetti voidaan tarkastaa seuraavalla Turing-koneella T_{par} :

S	A	$S \times A \times \{\rightarrow, \leftarrow, \downarrow\}$	S	A	$S \times A \times \{\rightarrow, \leftarrow, \downarrow\}$
s_0	0	$\langle s_0, 0, \rightarrow \rangle$	s_1	0	$\langle s_1, 0, \rightarrow \rangle$
s_0	1	$\langle s_1, 1, \rightarrow \rangle$	s_1	1	$\langle s_0, 1, \rightarrow \rangle$
s_0	$\sqcup$	$\langle k, \sqcup, \downarrow \rangle$	s_1	$\sqcup$	$\langle e, \sqcup, \downarrow \rangle$

Syötteellä "101" kone T_{par} laskee seuraavasti: $\langle s_0, \varepsilon, 101 \rangle \xrightarrow{T_{\text{par}}} \langle s_1, 1, 01 \rangle \xrightarrow{T_{\text{par}}} \langle s_1, 10, 1 \rangle \xrightarrow{T_{\text{par}}} \langle s_0, 101, \sqcup \rangle \xrightarrow{T_{\text{par}}} \langle k, 101, \sqcup \rangle$. ■

Epä-deterministisen laskennan piirteitä

- Vaihtoehtoista laskentaa muodostuu *laskentapuu*:



- Deterministisen koneen laskentapuuissa on vain yksi haara!

Määritelmä 8.7 (Epä-deterministinen) Turing-kone $T = \langle A, S, s_0, t \rangle$ **hyväksyy** kielen $L \subseteq (A \setminus \{\sqcup\})^*$, jos ja vain jos kaikille $x \in (A \setminus \{\sqcup\})^*$, $x \in L \iff$ koneella T on ainakin yksi hyväksyvä laskenta syötteellä x .

Epä-deterministiset Turing-koneet

- Tilansiirtofunktio t korvataan tilansiirtorelaatiolla $t : S \times A \rightarrow 2^{S \times A \times \{\rightarrow, \leftarrow, \downarrow\}}$.
- Tilasiirtymä konfiguraatiossa $\langle s, v, aw \rangle$ perustuu epä-deterministisesti valittuun kolmikkoon $\langle s', a', m \rangle \in t(s, a)$.
- Mahdollisia laskentoja voi olla useita.

Esimerkki 8.4 Olkoon $A = \{0, 1, \sqcup\}$ ja $S = \{s_0, k, e, p\}$. Määritellään epä-deterministinen Turing-kone seuraavasti:

S	A	$2^{S \times A \times \{\rightarrow, \leftarrow, \downarrow\}}$
s_0	$\sqcup$	$\{\langle s_0, 0, \rightarrow \rangle, \langle s_0, 1, \rightarrow \rangle, \langle p, \sqcup, \downarrow \rangle\}$

Yksi mahdollinen laskenta: $\langle s_0, \varepsilon, \sqcup \rangle \xrightarrow{T} \langle s_0, 1, \sqcup \rangle \xrightarrow{T} \langle s_0, 10, \sqcup \rangle \xrightarrow{T} \langle s_0, 101, \sqcup \rangle \xrightarrow{T} \langle p, 101, \sqcup \rangle$. ■

Päätösongelmien ratkaiseminen

- **Päätösongelmat** ovat ongelmia, joiden instanssien ratkaisuksi riittää yksinkertainen vastaus "kyllä" tai "ei".
- Päätösongelman ratkaiseminen Turing-koneella edellyttää ongelmatapausten esittämistä merkkijonoina ja Turing-koneen T laatimista "kyllä"-tapauksia vastaavan kielen ratkaisemiseen.

Esimerkki Onko 561 alkuluku?

Esimerkki 8.8 Lauselogiikan toteutuvuusongelmassa on tarkoituksena selvittää, onko syötteeksi annettu lause $\phi \in \mathcal{L}$ toteutuva vai ei. ■

$\implies$ Lauselogiikan toteutuvuusongelma voidaan samaistaa toteutuvien lauseiden joukkoa vastaavaan kieleen $\text{SAT} = \{\phi \mid \exists \mathcal{A} \subseteq \text{At}(\phi) : \mathcal{A} \models \phi\}$.

3. KESKEISET VAATIVUUSLUOKAT

- Päätösongelmien laskennallista vaativuutta voidaan analysoida asettamalla Turing-koneiden laskentaresursseille rajoituksia.

Määritelmä 8.9 Turing-kone $T = \langle A, S, s_0, t \rangle$ pysähtyy *polynomisessa ajassa* syötteen pituuden nähden, joss on olemassa aidosti kasvava polynomi p siten, että kaikille merkkijonoille $x \in (A \setminus \{\sqcup\})^*$, koneen T laskenta syötteellä x käsittää korkeintaan $p(|x|)$ konfiguraatiota.

Kaksi keskeistä kielten (ja päätösongelmien) luokkaa ovat:

1. Luokka **P** eli kielet, jotka voidaan ratkaista polynomisessa ajassa *deterministisellä* Turing-koneella.
2. Luokka **NP** eli kielet, jotka voidaan ratkaista polynomisessa ajassa *epädeterministisellä* Turing-koneella.

4. REDUSOITUVUUS JA VAIKEAT ONGELMAT

Määritelmä 8.11 Kieli $L_1 \subseteq (A_1 \setminus \{\sqcup\})^*$ on *reduoitavissa* kieleen $L_2 \subseteq (A_2 \setminus \{\sqcup\})^*$, jos ja vain jos löytyy syötteen $w \in (A_1 \setminus \{\sqcup\})^*$ pituuden suhteen polynomisessa ajassa deterministisellä Turing-koneella laskettava funktio $r : (A_1 \setminus \{\sqcup\})^* \rightarrow (A_2 \setminus \{\sqcup\})^*$ siten, että $w \in L_1 \iff r(w) \in L_2$.

Esimerkki 8.12 Olkoon $G = \langle S, K \rangle$ graafi, missä $K \subseteq S \times S$ antaa solmujen väliset kaaret. Kysymys graafin G kolmiväritettävyydestä, eli päätösongelma 3COL, voidaan *reduoida* lauselogiikan toteutuvuuskyseksi seuraavasti. Määritellään $r(G) = r(\langle S, K \rangle) = \{M_s \vee V_s \vee H_s \mid s \in S\} \cup \{\neg M_s \vee \neg M_t, \neg V_s \vee \neg V_t, \neg H_s \vee \neg H_t \mid \langle s, t \rangle \in K\}$. Nyt voidaan osoittaa, että $G \in 3COL \iff r(G) \in SAT$. ■

Päätösongelman SAT alustava luokittelu

Teoreema 8.10 Toteutuvuusongelma SAT kuuluu luokkaan **NP**.

Todistuksen idea: Voidaan laatia epädeterministinen Turing-kone T_{sat} , jolle annetaan syötteen lause $\phi \in \mathcal{L}$ ja joka

1. valitsee epädeterministisesti totuusjakelun $\mathcal{A} \subseteq \text{At}(\phi)$,
2. laskee lauseen ϕ totuusarvon totuusjakelussa $\mathcal{A}$ ja
3. pysähtyy tilaan k , jos $\mathcal{A} \models \phi$, ja muutoin tilaan e .

Tarvittava laskenta pystytään suorittamaan polynomisessa ajassa lauseen ϕ merkkijonoesityksen pituuden suhteen.

Voidaan osoittaa, että $\phi \in SAT$, jos ja vain jos koneella T_{sat} on ainakin yksi hyväksyvä laskenta syötteellä ϕ . □

Vaikeat ja täydelliset ongelmat

Määritelmä 8.13 Olkoon **C** jokin kielten luokka. Kieli L on

1. **C-vaikea**, jos ja vain jos jokainen luokan **C** kieli L' voidaan redusoida kieleen L polynomisessa ajassa ja
2. **C-täydellinen**, jos ja vain jos L kuuluu luokkaan **C** ja L on **C-vaikea**.

Teoreema 8.14 Toteutuvuusongelma SAT on **NP-vaikea**.

Muutama keskeinen huomio:

- **C-täydelliset** ongelmat ovat vaativimpia luokan **C** ongelmia.
- SAT on **NP-täydellinen** ongelma.

DPLL-algoritmi

Olkoon S äärellinen klausuulijoukko ja L joukko literaaleja.

Function OnToteutuva(S, L): **Boolean**;

Begin

$\langle S, L \rangle := \text{Yksinkertaista}(S, L);$

If $\square \in S$ **Then Return False**;

If $S = \emptyset$ **Then Return True**;

$I := \text{Valitse}(S, L);$

If OnToteutuva($S, L \cup \{I\}$) **Then Return True**

Else Return OnToteutuva($S, L \cup \{\bar{I}\}$);

End

Tehokkaita DPLL-toteutuksia ja benchmark-ongelmia on saatavilla verkossa (kts. esim. <http://www.satlive.org>).

PÄIVÄN PÄHKINÄ

Esimerkin 8.15 DPLL-simulaatiossa oletettiin literaalit P_1 ja V_1 , jolloin voitiin päätellä kartanväritysesimerkin klausuulijoukosta S literaalit

$$P_1, \neg V_1, \neg S_1, \neg P_2, \neg P_3, \neg P_4, V_2, \neg S_2, \neg V_3, \neg V_4, S_3, S_4,$$

jolloin päädyttiin ristiriitaan.

- Osoita, että $S \models \neg P_1 \vee \neg V_2$.
- Mieti kytkentöjä DPLL-algoritmin suorituksen aikana syntyvän laskentapuun ja resoluutiomenetelmän täydellisyydestä yhteydessä käytetyn puurakenteen välillä.
- Voidaanko DPLL-algoritmin suorittamaa laskentaa hyödyntää klausuulin $\neg P_1 \vee \neg V_1$ johtamiseen joukosta S ?

TAVOITTEET

- Olet kerrannut Turing-koneisiin liittyvät keskeiset määritelmät.
- Osaat määritellä luokat **P** ja **NP** ja selittää niiden välisen eron määritelmällisestä näkökulmasta ($\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ on avoin kysymys).
- Tiedät, mitä päätösongelman **NP**-täydellisyys tarkoittaa ja mitä **NP**-täydellisyyden osoittaminen vaatii.
- Ymmärrät reduktioiden roolin päätösongelmien välisinä muunnoksia ja tunnet tällaisista muunnoksista joitain esimerkkejä.
- Osaat simuloida DPLL-algoritmia annetulle klausuulijoukolle.
- Olet kokeillut jotain toteutuvuustarkistinta käytännössä.