



## Pätevyys ja toteutuvuus

- CTL/CTL\*

Lause (tilalause)  $P$  on pätevä mallissa  $\mathcal{M}$  ( $\mathcal{M} \models P$ ), jos jokaiselle mallin tilalle  $s$ ,  $\mathcal{M}, s \models P$ .

Lause  $P$  on toteutuva, jos on olemassa malli  $\mathcal{M}$  ja tila  $s$  siten, että  $\mathcal{M}, s \models P$ .

- LTL

Lause (polkulause)  $P$  on pätevä mallissa  $\mathcal{M}$  ( $\mathcal{M} \models P$ ), jos jokaiselle mallin täydelle polulle  $x$ ,  $\mathcal{M}, x \models P$ .

Lause  $P$  on toteutuva, jos on olemassa malli  $\mathcal{M}$  ja sen täysi polku  $x$  siten, että  $\mathcal{M}, x \models P$ .

- Lause on pätevä, jos se on pätevä jokaisessa mallissa.

Huom! Lause on pätevä, joss sen negaatio ei ole toteutuva.



- Mallintarkastuksen kannalta vastavuus:

CTL:

$$\mathcal{M}, s_0 \models P$$

LTL:

$$\mathcal{M}, x \models P$$

kaikilla täysillä poluilla

$$x = (s_0, \dots).$$

- CTL- ja LTL-operaattorit samantyyppisiä mutta tulkinnat eroavat!

- Mahdollisuuslauseita ei voida ilmaista LTL:llä.

**Esimerkki.** CTL-lauseelle **AGEFP** ei löydy vastaavaa LTL-lauseetta.

Tarkastellaan LTL-lauseetta **GFP**.

Lause **AGEFP** pätevä mallissa  $\mathcal{M} = \langle S, R, v \rangle$ :

$$S = \{s_0, s_1\}, R = \{\langle s_0, s_0 \rangle, \langle s_0, s_1 \rangle, \langle s_1, s_0 \rangle\},$$

$$v(s_0, P) = \text{false}, v(s_1, P) = \text{true}$$

mutta LTL-lause **GFP** ei ole pätevä: se on epätosi täydellä polulla  $(s_0, s_0, s_0, \dots)$



## CTL vs. LTL

- LTL-lauseen totuus määräytyy mallin antamien **täysien polkujen** perusteella.

- CTL-lauseen totuus määräytyy mallin  $\mathcal{M} = \langle S, R, v \rangle$  antaman **laskentapuun**  $\hat{\mathcal{M}} = \langle \hat{S}, \hat{R}, \hat{v} \rangle$  perusteella.

Teknisesti malli  $\mathcal{M}$  voidaan "avata" laskentapuuksi  $\hat{\mathcal{M}} = \langle \hat{S}, \hat{R}, \hat{v} \rangle$

(i) ottamalla solmuiksi  $\hat{S}$  parit  $\langle s, n \rangle$ , missä  $s \in S$  ja  $n$  luonnollinen luku, joka antaa kullekin puun tilalle yksikäsitteisen tunnusteen,

(ii) aloittamalla tilasta  $\langle s_0, 0 \rangle$  ja

(iii) avaamalla mallia käyttämällä sääntöä:

jos  $\langle s, n \rangle \in \hat{S}$  ja  $sRt$ , niin  $\langle t, m \rangle \in \hat{S}$  ja  $\langle \langle s, n \rangle, \langle t, m \rangle \rangle \in \hat{R}$ , missä  $m$  on uusi luonnollinen luku, joka ei ole muualla käytössä.

Valuaatio:  $\hat{v}(\langle s, n \rangle, P) = v(s, P)$ .



- Siis "on olemassa polku" -tyyppiset CTL-lauseet eivät ole ilmaistavissa LTL-lauseilla.

**Esimerkki.** CTL-lauseelle **EFP** ei löydy vastaavaa LTL-lauseetta.

Tarkastellaan LTL-lauseetta **FP**.

Lause ei ole pätevä y.o. mallissa  $\mathcal{M}$ : epätosi täydellä polulla

$$(s_0, s_0, s_0, \dots)$$

mutta lause **EFP** on pätevä.

- Reiluusominaisuudet eivät ole ilmaistavissa CTL-lauseina.

**Esimerkki.** LTL-lauseelle **FGQ** ei löydy vastaavaa CTL-lauseetta.

Tarkastellaan edellä olevaa mallia  $\mathcal{M}$ , jossa

$$v(s_0, Q) = \text{true}, v(s_1, Q) = \text{false}$$

Lause **FGQ** on toteutuva (täysi polku  $(s_0, s_0, \dots)$ )

mutta CTL-lause **AFAGQ** ei ole eikä myöskään lause **EFAGQ**.

## Esimerkkejä

- **EF**(*started*  $\wedge$   $\neg$ *ready*)  
On mahdollista päästä tilaan, jossa *started* tosi muttei *ready*.
- **AG**(*req*  $\rightarrow$  **AFack**)  
Jos pyyntö tulee, saadaan siihen kuittaus.
- **AGAF***enabled*  
*enabled* on tosi äärettömän usein jokaisella laskentapolulla
- **AGEF***restart*  
Jokaisesta tilasta on mahdollista päästä *restart* tilaan

## Saavutettavuusominaisuudet

- Yksinkertaisin ominaisuusluokka, joka ilmaisee, että jokin järjestelmän tila (jossa annetut ehdot *P* ovat voimassa) on saavutettavissa (järjestelmän alkutilasta).
- Vastaava temporaalilause: **EFP**.
- Esimerkkejä:  
**EF**(*started*  $\wedge$   $\neg$ *ready*)  
**EF**(*restart*)
- Monimutkaisempi luokka: ehdollinen saavutettavuus **E**(*QUP*)
- Esimerkki: **E**( $\neg$ *restart***U***ready*)

## Vaatimusmäärittelyt

- Temporaalilogiikkaa käytetään reaktiivisten järjestelmien vaatimusmäärittelyyn
- Tyypilliset järjestelmät vaatimukset voidaan jakaa seuraaviin luokkiin
  - Saavutettavuusominaisuudet
  - Turvallisuusominaisuudet
  - Elävyyso ominaisuudet
  - Reiluusominaisuudet

## Turvallisuusominaisuudet

- Turvallisuusominaisuudet ilmaisevat, ettei mitään pahaa ei tapahdu.
- Turvallisuusominaisuudella tarkoitetaan vaatimusta, jolle löytyy aina "äärellinen vastasuoritus":  
jos järjestelmä ei toteuta annettua turvallisuusominaisuutta, on sillä äärellinen suoritus, joka rikkoo omaisuuden.
- Esimerkkejä
  - Keskinäinen poissulkeminen: **AG** $\neg$ (*atCS*<sub>1</sub>  $\wedge$  *atCS*<sub>2</sub>)
  - Osittainen oikeellisuus: *at*<sub>l<sub>0</sub></sub>  $\wedge$  *P*  $\rightarrow$  **AG**(*at*<sub>l<sub>h</sub></sub>  $\rightarrow$  *Q*)

## Elävyyssominaisuudet

- Elävyyssominaisuudet ilmaisevat, että jotain hyvää tapahtuu.
- Elävyyssominaisuudella ei ole "äärellistä vastasuoritusta": jos järjestelmä ei toteuta annettua elävyyssominaisuutta, tämä näkyy vain äärettömissä suorituksissa.
- Esimerkkejä
  - (Toistettava) saavutettavuus:  $\mathbf{AGF}restart$
  - Temporaali-implikaatio:  $\mathbf{AG}(P \rightarrow \mathbf{AF}Q)$
  - Nälkiintymättömyys:  $\mathbf{AG}(atTry_i \rightarrow \mathbf{AF}atCS_i)$
  - Totaalinen oikeellisuus:  $atl_0 \wedge P \rightarrow \mathbf{AF}(atL_h \wedge Q)$

## Reiluusominaisuudet ja CTL

- Käytettäessä CTL-logiikkaa reiluusvaatimukset käsitellään muuttamalla polkukvanttorien ( $\mathbf{A}/\mathbf{E}$ ) semantiikkaa: kvantifiointi on yli reilujen polkujen (eikä kaikkein polkujen kuten perustapauksessa).
- Reilusehdot annetaan joukkona lauseita  $F$ .
- Täysi polku  $x$  on  **$F$ -reilu** joss jokaiselle  $P \in F$  jokin tila, jossa  $P$  on tosi, esiintyy äärettömän usein polulla  $x$ .

## Reiluusominaisuudet

- Reiluusominaisuudet ovat elävyyssominaisuuksia, jotka vaativat, että annetun ehdon toteuttava tila toistuu äärettömän usein.
- Reilusehdot eivät ole suoraan ilmaistavissa CTL-logiikalla mutta ovat LTL-logiikalla.
- Esimerkkejä
  - Määritellään prosessille atomilauseet *en* (prosessi on toimintavalmis) ja *ex* (prosessi suoritetaan).
  - Ehdoton reiluus:  $\mathbf{GF}ex$
  - Heikko reiluus:  $\mathbf{F}Gen \rightarrow \mathbf{GF}ex$
  - Vahva reiluus:  $\mathbf{GF}en \rightarrow \mathbf{GF}ex$

## Reiluusominaisuudet ja CTL

- $F$ -reilu semantiikka ( $\models_F$ ) siis huomioi ainoastaan  $F$ -reilut polut.
- Relaatio  $\models_F$  määritellään kuten  $\models$  paitsi, että polkukvantifiointi koskee vain  $F$ -reiluja polkuja:
  - $\mathcal{M}, s \models_F P$  joss on olemassa tilasta  $s$  alkava  **$F$ -reilu** täysi polku ja  $v(s, P) = \text{true}$ , kun  $P$  on atomilause
  - $\mathcal{M}, s \models_F \mathbf{A}(PUQ)$  joss mallissa  $\mathcal{M}$  kaikille  **$F$ -reiluille** täysille poluille  $(s_0, s_1, \dots)$  missä  $s = s_0$ , on olemassa  $i$ , jolle  $\mathcal{M}, s_i \models_F Q$  ja kaikille  $j < i$ ,  $\mathcal{M}, s_j \models_F P$ .
  - $\mathcal{M}, s \models_F \mathbf{E}(PUQ)$  joss mallissa  $\mathcal{M}$  on olemassa  **$F$ -reilu** täysi polku  $(s_0, s_1, \dots)$  siten, että  $s = s_0$  ja on olemassa  $i$ , jolle  $\mathcal{M}, s_i \models_F Q$  ja kaikille  $j < i$ ,  $\mathcal{M}, s_j \models_F P$ .
- Esim. ehdoton reiluus:  $F = \{ex\}$  ja reilu kanava:  $F = \{send \rightarrow rec\}$