מבוא למערכות מבוזרות

אורן דנון

שיעור 1 – מבוא

1.1 מהו תכנות מבוזר?

מערכת מבוזרת היא אוסף של התקני מיחשוב היכולים לתקשר אחד עם השני. בפרט נציין כי חישוב מקבלי הינו מקרה פרטי של חישוב מבוזר בו כל הצדדים שואפים להשיג מטרה אחת, אולם במערכת מבוזרת אין הכרח לכך, ולכל אחד יכול להיות מטרות שונות. מערכות מבוזרות מאפשרות לנו לשתף מידע בין ישויות שונות, לטפל בכמויות מידע רבות, למקבל חישוב על פני מכונות רבות, לבנות מערכות הפרוסות במרחקים מרובים, בנוסף באמצעות מערכת מבוזרת נוכל להשיג עמידות לתקלות.

1.2 מדוע מערכות מבוזרות שונות?

במערכות מבוזרות צצים אתגרים חדשים כגון:

- **סינכרון** – קשה לתאם בין התקני מיחשוב שכן המחזור של כל התקן שונה, וקשה לתאם בין ההתקני (יחידות החישוב). אין זמן גלובלי שכולם יכולים לעקוב אחריו.
- איך להסכים על סדר פעולות חישוב (לא בהכרח לכולם יש אותה מטרה).
- ישנם עיכובים בלתי צפויים, לפעמים לא כל ההודעות מגיעות, לפעמים מחשב מסוים לא יגיב (בין אם כי הוא מכובה ובין אם האקר ישתלט על המערכת).

רוב הקורס יעסוק בנכונות, שכן אנו נראה כי ישנו קושי רב להשיג מערכת מבוזרת שעונה על הדרישות שנגדיר, לפיכך רוב הקורס לא יעסוק ביעילות.

1.3 מודלים של מערכות מבוזרות

נתמקד בשני דרכים לפשט מערכת מבוזרת:

1. **מערכת העברת הודעות:**
   - קודקודים/תהליכים מתקשרים באמצעות החלפת הודעות.
   - קשת מכוונת מקודקוד v לקודקוד u מסמלת שיש לקודקוד v את היכולת לשלוח הודעה לקודקוד u. (אם הגרף לא מכוון הכוונה היא שיש קשת מכוונת בשני הכיוונים).
2. **זיכרון משותף:**

- תהליכים מתקשרים ע"י כתיבה וקריאה מזכרון משותף.

במהלך הקורס נחליף בין המודלים כאוות נפשנו כיוון שיש בידעתינו כיצד לסמלץ ריצה במודל אחד, באמצעות המודל השני.

1.4 סינכרון

- **מערכות סינכרוניות:** ישנו שעון שעובד בפולסים, שהיינו זמן דיסקרטי, הסיבוב ה–i הוא הזמן בין הזמן $i-1$ לזמן ה–i.
- **סינכרון במערכת העברת הודעות:** הסיבוב ה-i בתחילת הסיבוב (כלומר בזמן $i-1$) כל תהליך שולח הודעות, ההודעות נשלחות ומעובדות בזמן i.
- **סינכרון בזיכרון משותף:** בכל סיבוב כל תהליך יכול לגשת לתא אחד בזיכרון.

1.4.1 מערכת אסינכרונית

נניח בהלך הקורס שמהירות העיבוד ועיכוב ההודעות סופיים אחרת נקבל מערכת שלא ניתנת לחיזוי. בנוסף נניח שזמן העיבוד/זמן עיכוב ההודעה נקבע במקרה הגרוע ע"י מתזמן עוין.

1. **שליחת הודעות:**
   - הודעות תמיד נשלחות (בריצה ללא תקלות).
   - עיכוב ההודעות הינו שירורותי (נקבע ע"י היריב (המתזמן העוין)).
2. **זיכרון משותף:**
   - כל התהליכים עושים לבסוף את הצעד הבא (בריצה ללא תקלות).
   - מהירות התהליכים היינה שירורותית (נבחרת ע"י היריב).

ישנם מודלים שבספקטרום בין מערכת סינכורניות למערכת א־סינכורנית אולם לא נעסוק בהם בקורס זה.

1.5 תקלות/נפילות

- **כשל התרסקות:** צומת מפסיק לעבוד בנקודה מסוימת של הריצה.
- **כשל ביזנטי:** קודקוד מתנהג בצורה שירותית לחלוטין. קודקודים ביזנטים שונים יכולים לקשור קשר (להתאחד).
- **כשל השמטה:** קודקוד/תהליך/ערוץ תקשורת מפסיק לעבוד ***זמנית***. לדוגמה: חלק מההודעות הולכות לאיבוד.
- **חסינות** – Resilience – מספר התקלות שהמערכת יודעת להתמודד איתה, כלומר לכמה כשלונות המערכת חסינה.

1.6 נכונות של מערכת מבוזרת

נגדיר מספר קירטריונים שיסיעו להעריך עד כמה המערכת עונה ומספקת את הדרישות מבחינת נכונות, כלומר עושה את מה שאנו מצפים ממנה.

- **בטיחות**–Safety – שום דבר רע לא קורה.
- **חיות**–Liveness – משהו טוב מתבצע.
- **הגינות**–Fairness – משהו טוב מתבצע לכולם.

1.7 תיאור פורמלי של העברת הודעות

כפי שציינו המערכת מורכבת מ–n קודקודים (דטרמיניסטים), וקבוצת קשתות בין זוגות של קודקודים, כך שבכל זמן, לכל קודקוד v_i יש מצב המאפיין אותו Q_i (המצב הפנימי יכול להכיל קלטים, משתנים מקומיים, שעונים, חישובי עזר וכיו"ב. בנוסף הוא מכיל את כל ההיסטוריה של האירועים שניצפו).

1.7.1 סוגי אירועים

- **שליחת אירוע**–Send Event קודקוד v_i שם הודעה בערוץ התקשורת לקודקוד כלשהו v_j.
- **קבלת אירוע**–Receive Event קודקוד v_i קיבל הודעה - חייב להיות עוקב ע"י שליחת אירוע.
- **תזמון אירוע**–Timing Event–אירוע המופעל בצומת לפי שעון מקומי.

הערה: אירועים יכולים להפעיל חישובים מקומיים שיכולים להפעיל אירועים אחרים.

1.8 קונפיגורציה

קונפיגורציה C: היא ווקטור של n המצבים (בזמן נתון). קונפיגורציה = מצב המערכת.

1.9 סבב הרצה — Execution Fragment

סדרה של קונפיגורציות ואירועים מתחלפים. **לדוגמה:** $C_0, \Phi_1, C_1, \Phi_2, C_2, \Phi_3, \ldots$ כאשר ה–C_i–ים הם קונפיגורציות וה–Φ_i הם אירועים. (כמובן שכל שלשה C_{i-1}, Φ_i, C_i צריכה להיות עקבית עם כלל השינוי, כלומר Φ_i משפיעה **רק** על המצב שקיבל את האירוע.

1.10 הגדרות נוספות

- **הרצה:** הרצה היא שבר הרצה המתחיל מהקונפיגורציה ההתחלתית, כלומר C_0.
- **מתזמן:** הרצה בלי הקונפיגורציות, אבל עם הקלטים (סדרת האירועים של ריצה והקלטים).
- **מצב מקומי:** מצב של קודקוד v_i שלא כולל את המצב של ההודעות שנשלחו ע"י v_i (כלומר לא יודע אם ההודעה הגיעה או אבדה).

- **יריב:** כל עוד הוא מכבד את הכללים של המודל, הוא יכול לעשות מה שהוא רוצה (בפרט הוא יעשה מה שהכי טוב לו, כלומר מה שהכי רע לנו).

נניח שהקודקודים דטרמיניסטים, פרט לשיעור יחיד בוא נתיר לקודקוד בודד להטיל מטבע.

Schedule Restriction & indistinguishability

- Schedule Restriction — **בהינתן מתזמן S נגדיר את $S|i$ בתור תת–הסדרה של S המורכבת הקלט של v_i וכל האירועים המתרחשים בקודקוד v_i. דוגמה: יהי v_1, v_2, v_3, נסמן ב–$s_{i\to j}$ או ב–s_{ij} שליחת אירוע מ–i ל–j ונסמן ב– r_{ij} את האירוע ש–i קיבל הודעה מ–j. נתונה המתזמן:**

$$S = s_{1\to3}, s_{2\to3}, s_{3\to1}, r_{13}, s_{3\to2}, r_{23}, s_{1\to3}, s_{2\to1}, r_{31}, r_{12}, r_{32}$$

$S|1 = s_{1\to3}, r_{13}, s_{1\to3}, r_{12}$.

- Indistinguishability — **יהי S ו–S' שני תזמונים ויהי v_i קודקוד. נניח ש–$S|i = S'|i$ ושל–v_i יש קלט מסויים ב–S ו–S'. אזי v_i יבצע את אותה פעולה ב–S וב–S'.**
הוכחה: המצב של קודקוד v_i תלוי אך ורק בקלט של v_i וב–$S|i$, עבור קודקודים דטרמיניסטים, הפעולה הבאה תלויה אך ורק במצב הנוכחי, כיוון שהמצב הנוכחי אותו דבר בשני המקרים הפעולה תיהיה זהה.

מתזמן יקרא **תקין** אם:

1. ישנם אינסוף צעדי חישוב עבור כל קודקוד.
2. כל הודעה נשלחת לבסוף.

נשים לב שהתנאים אלו הם תאים להוגנות כלומר fairness.

שיעור שני — בעיית שני הגנרלים

2.1 הגדרת הבעיה

- **המודל:**
  - שני קודקודים הפועלים בצורה דטרמיניסטית
  - תקשורת סינכרונית
  - ההודעות בלתי אמינות (הודעות יכולות להאבד)
- **הקלט:** קודקוד מתחיל עם אחד משני קלטים אפשריים 0 או 1.
- **פלט:** כל קודקוד צריך לפלוט/להחליט 0 או 1
- **הסכמה:** שניהם חייבים להוציא את אותו פלט, כלומר להסכים על הפלט.
- Validity: אם לשני הקודקודים אותו קלט $x \in \{0,1\}$ ושום הודעה לא אבדה, אזי שניהם מוציאים/מחליטים x.
- **סיום** - Termination – התוכנית מסתיימת לאחר מספר סיבובים סופי (כלומר חסום).

הבנת הבעייה

- Validity – נועד למנוע את המקרה הטריוואלי ששני הקודקודים (הגנרלים) מחליטים תמיד להוציא 1 או תמיד להוציא אפס.
- **הסכמה** היינה הדרישה העיקרית של הבעיה
- הבעיה נקראת בעיית שני הגנרלים שכן הבעיה שקולה לבעיה בה לשני גנרלים שצריכים להחליט על זמן מתי לתקוף את האויב, על מנת לנצח (אחרת שניהם מפסידים). הם מתקשרים דרך שליחים שיכולים להירצח ע"י האויב. והבעיה היא להסכים מתי לתקוף את האויב.

הבעיה הפשוטה שהצגנו לא פתירה, כלומר לא קיים אלגוריתם הפותר אותה. הוכחה מתבססת על משפט ה– Indisinguishability שהוצג בשיעור הקודם. בפרט בהוכחה נעזר ב–validity.

2.1.1 רעיון ההוכחה

1. נניח בשלילה שיש אלגוריתם כנ"ל, אזי לאחר זמן סופי, האלגוריתם עוצר ושני הקודקודים מחליטים על פלט, כלומר קיים סיבוב אחרון $T \in \mathbb{N}$.
2. נבנה סדרת הרצות $E_1, E_2, E_3, ..., E_k$ שיקיימו:
   - עבור כל $i \in \{1, 2, ..., k\}$ יתקיים $E_{i-1} \underset{v_1}{\sim} E_i$ או $E_{i-1} \underset{v_2}{\sim} E_i$
   - ב־E_0 הפלט צריך להיות 0 ובהרצה E_k הפלט צריך להיות 1 סתירה.

2.1.2 ההוכחה

נניח בשלילה שקיים אלגוריתם כנ"ל, נניח שהוא מסתיים לאחר T סיבובים, ונבנה סדרת הרצות דומות באופן הבא:

- תהי E_0 ההרצה בה הקלט לשני הקודקודים היינו 0 ושום הודעה לא אבדה.
- תהי E_1 ההרצה בה אבדה הודעה בסיבוב ה־T בה"כ נניח כי ההודעה ש־v_1 שולח ל־v_2 נאבדה, לפיכך בעיני v_1 ההרצות E_0 ו־E_1 בלתי ניתנות להבדלה (מעתה זהות), כלומר מתקיים $E_0 \underset{v_1}{\sim} E_1$
- תהי E_2 ההרצה בה בסיבוב ה־T שני ההודעות הלכו לאיבוד, כלומר מתקיים $E_1 \underset{v_2}{\sim} E_2$
- E_{2i} $\forall_{1\leq i<T}$ שני ההודעות בסיבוב ה־ $T+1-i$ אבדו, וכל השאר כמו בסיבוב הקודם.
- E_{2i+1} $\forall_{1\leq i<T}$ אחת מההודעות בסיבוב ה־$T-i$ אבדה.
- E_{2T} הקלט עבור שני הקודקודים היינו 0, ושום הודעה לא עברה בהצלחה. שני הפלטים הם אפס עקב ה־דמיון.
- E_{2T+1} הקלט של הקודקוד v_1 הוא אחד, אבל הקלט של קודקוד v_2 נשאר 0, שום הודעה לא עוברת. (דומה כי v_1 יודע שהוא ישתנה אומנם v_2 לא יכול להבחין בהבדל בקלט של v_1 לפיכך דומה לפי v_2).
- E_{2T+2} שני הקלטים הם 1 שום הודעה לא עוברת.
- $E_{2T+2i+1}$ $\forall_{0\leq i\leq T}$ כל ההודעות עד הסיבוב ה־i מגיעות ואחד מההודעות בסיבוב ה־i מגיעה, וכל שאר ההודעות בסיבובים הגדולים מ־i לא מגיעות.
- $E_{2T+2i+2}$ $\forall_{0\leq i\leq T}$ כל ההודעות עד הסיבוב ה־i מגיעות (כולל) וכל השאר לא מגיעות.
- E_{4T+2} כל ההודעות מגיעות ושני הקלטים אחד אבל הפלט הוא 0 בגלל הדמיון בסתירה ל־Validity.■

2.1.3 סיכום ההוכחה

- התחלנו מהרצה בה שני הקודקודים קבלו את הקלט 0 ושום הודעה לא נאבדה, ע"פ Validity שניהם צריכים להחליט 0.
- הורדנו הודעות אחת בכל הרצה בצורה כזאת שתשמר דמיון בין ההרצות.
- כשהגענו להרצה בה שום הודעה לא עברה, החלפנו את הקלט של אחד מהקודקודים ובהרצה הבאה של השני.
- החזרנו את ההודעות בצורה שתשמר את הדמיון בין ההרצות עד שחזרנו להרצה בה כל ההודעות התקבלו.
- לפי הדמיון הפלט 0 אומנם לפי Validity הפלט אמור להיות 1, כי שני הקלטים בההרצה האחרונה הם 1, סתירה.

נעיר כעת שלא קשה להכליל את הבעיה למקרה בו $n \in \mathbb{N}$ כלשהו.

2.2 בעיית הגנרלים: אלגוריתם הסתברותי

ראינו שלא ניתן לפתור את בעיית הגנרלים כאשר הקודקודים דטרמיניסטים, כעת נסקור מה קורה כאשר אנו מתירים מקריות.
מסתבר כי בעיית הגנרלים יכולה להיפתר אם:

- אנו מרשים לאחד מהגנרלים להטיל מטבעות.
- אנו מסתפקים בכך שהסכמה מושגת בהסתברות $1 - \varepsilon$ (עבור אפסילון קטן דיו).

לפני שנראה אלגוריתם שפותר את הבעיה ונוכיח נכונות נתבונן בפונקציית העזר הבאה:

2.2.1 אלגוריתם הרמות — The Level Algorithm

תכונות האלגוריתם

האלגוריתם מספק את התכונות הבאות:

- שני הקודקודים מחשבים רמה.
- בסוף, ההפרש בין הרמות היינו אחד לכל היותר.
- הרמות לבסוף מודדות את מספר ההתמסרויות (שידורים/מסירות) היוצאות והנכנסות.

האלגוריתם

1. שני הרמות מאותחלות ל–0.
2. בכל סיבוב שני הקודקודים שולחים את הרמה הנוכחית אחד לשני.
3. כאשר קודקוד u ברמה ℓ_u מקבל הודעה מקודקוד v שמציין שרמתו ℓ_v, u מעדכן את רמתו ל–$\ell_u = \max\{\ell_u, \ell_v + 1\}$

הבחנות, טענות, ומשפטים

הבחנה: הרמה של קודקוד אף פעם לא יורדת.
משפט: בכל רגע, הרמות נבדלות באחד לכל היותר.

הוכחה: באינדוקציה על מספר הסיבובים.

- **בסיס:** בסיבוב הראשון $\ell_v = \ell_u = 0$
- נניח שהטענה נכונה לאחר t סיבובים ונראה עבור $t+1$.
- אם $\ell_u = \ell_v$ כאשר אנו בסיבוב ה-t הטענה נכונה שכן הרמה לעולם לא יורדת, ועולה לכל היותר ב-1.
- נניח בלי הגבלת הכלליות ש- $\ell_u = \ell_v + 1$. בסיבוב ה-t, כיוון ש-ℓ_u לא משתנה ו-ℓ_v גדל לכל היותר באחד, הטענה נכונה. ■

משפט: אם כל ההודעות נמסרו, אזי שני הרמות שוות למספר הסיבובים.

הוכחה: נוכיח באינדוקציה על מספר הסיבובים:

- **בסיס:** עבור ההתחלה הטענה נכונה, ע"פ האתחול.
- כעת נניח שהטענה נכונה, כלומר $\ell_u = \ell_v = t$, עבור $t \in \mathbb{N}$ ונשקול את הסיבוב $t+1$.
- **צעד האינדוקציה:** יהי ℓ'_u ובהתאמה ℓ'_v הרמה של קודקוד u והרמה של קודקוד v בסיבוב ה- $t+1$, ע"פ הגדרת האלגוריתם שני הקודקודים שולחים את הרמה שלהם לקודקוד השני, וע"פ ההנחה שני ההודעות מתקבלות, לפיכך מתקיים ש- $\ell'_u = \ell_u + 1 = t+1$ ומתקיים $\ell'_v = \ell_v + 1 = t+1$ ■

משפט: הרמה ℓ_u של קודקוד u שווה לאפס אם ורק אם u לא מקבל שום הודעה.

הוכחה: נזכיר שמתקיים:

$$\left[\mathrm{A} \iff \mathrm{B}\right] \iff \left[(\mathrm{B} \implies \mathrm{A}) \wedge (\neg\mathrm{B} \implies \neg\mathrm{A})\right]$$

- $A \implies B$ אם u לא מקבל שום הודעה, ע"פ האלגוריתם u לא מעדכן את הרמה שלו מהאיתחול, לפיכך רמתו שווה ל-0.
- $\neg\mathrm{A} \implies \neg\mathrm{B}$ נניח כעת כי קודקוד u מקבל הודעה ונראה כי $\ell_u \neq 0$, בפעם הראשונה ש-u מקבל הודעה הוא מעדכן את רמתו לאחד, וע"פ ההבחנה שהרמה לא יורדת לעולם הטענה נובעת. ■

סיכום אלגוריתם הרמות

אם האלגוריתם רץ $r \in \mathbb{N}$ סיבובים אזי:

1. בסוף, שני הרמות נבדלות באחד לכל היותר.
2. אם כל ההודעות נמסרו, אזי שני הרמות שוות ל־r.
3. הרמה של קודקוד u היא לפחות 1 אם ורק אם u מקבל לפחות הודעה אחת.

2.2.2 האלגוריתם האקראי של שני גנרלים

הנחות

- נניח ש־u יכול להשתמש באקראיות(לא נניח אותה הנחה על v כלומר רק קודקוד u משתמש באקראיות).
- נניח שהקלטים האפשריים הם 0 ו־1.

האלגוריתם

1. קודקוד u בוחר מספר $t \in \{1, 2, ..., r\}$ בצורה אחידה(r יפורט מאוחר יותר).
2. שני הקודקודים מרצים את אלגוריתם הרמות עבור r סיבובים. בזמן הרצת אלגוריתם הרמות שני הקודקודים כוללים את הקלטים שלהם בכל הודעה וקודקוד u כולל גם את ערכו של t.
3. בסוף, קודקוד מחליט 1 אם:

   ג.1 הקודקוד יודע את t והוא ראה את שני הקלטים,

   ג.2 שני הקלטים שווים לאחד ו־

   ג.3 הרמה של הקודקוד היא לפחות t
4. אחרת הקודקוד מחליט 0.

משפט: אם לפחות אחד המקלטים 0, אזי שני הקודקודים מחליטים 0.

הוכחה: נובע מתיאור האלגוריתם, בפרט לא מקיים **ג.2** ■

משפט: נניח שני הקלטים הם 1

1. אם בנוסף שום הודעה לא נאבדת, אזי שני הקודקודים מחליטים 1.
2. שני הקודקודים פולטים אותו ערך, אלא אם כן $\{\ell_u, \ell_v\} = \{t-1, t\}$

הוכחה: נשקול את המקרים הבאים:

- נניח תחילה כי $\{\ell_u, \ell_v\} = \{t-1, t\}$. נניח ש־$\ell_u = t-1, \ell_v = t$ (המקרה המשלים דומה). ע"פ **ג.3** קודקוד u מחליט 0. כיוון ש־$\ell_v = t > 0$, v יודע את t ואת שני הקלטים, בנוסף ע"פ ההנחה של המשפט מתקיים **ג.2** ולכן v מחליט 1.

- כעת נניח $\{\ell_u, \ell_v\} \neq \{t-1, t\}$. אם $\ell_u < t$ וגם $\ell_v < t$, אזי שניהם מחליטים 0 ע"פ **ג.3**. אחרת מתקיים $\ell_u \geq t$ וגם $\ell_v \geq t$, שכן הרמה שונה לכל היותר באחד, אבל עבור אף אחד לא ברמה $t-1$(כלומר שניהם מספקים את **ג.3**). כיוון כיוון ש־$t > 0$, שניהם יודעים את t ושניהם ראו את הקלטים (כלומר מספק את **ג.1** ואת **ג.2**). ולכן מחליטים 1.

- לסיום, אם שום הודעה לא אבדה, אזי $\ell_u = \ell_v = r \geq t$ ולפיכך, ע"פ הנאמר לעיל שניהם פולטים 1. ■

משפט: האלגוריתם משיג הסכמה עם הסתברות של לפחות $1 - \frac{1}{r}$

הוכחה: נזכיר כי שני רמות נבדלות אחת מהשני לכל היותר ב־1.

- כיוון שהרמות תלויות אך ורק במספר ההודעות שהתקבלו, היריב יבחר את המקרה הגרוע עבורנו, דהיינו: $\{\ell_u, \ell_v\} = \{i-1, i\}$ עבור $i \in \{1, ..., r\}$ כלשהו.

- כיוון שהיריב אינו יודע את t אזי ההסתברות למאורע בו היריב בוחר את $\{\ell_u, \ell_v\} = \{t-1, t\}$ היא $\frac{1}{r}$.

- ע"פ המשפט הקודם, הקודקודים מגיעים להסכמה בכל מקרה אחר, לפיכך ההסתברות לכישלון היא $\frac{1}{r}$ ■

2.2.3 חסם תחתון על השגיאה האקראית

באמצעות שיטות דומות לשיטות שראינו בהוכחת היעדר הפתרון לבעיית שני הגנרלים במודל הדטרמיניסטי, נוכל להוכיח חסם תחתון על השגיאה.

גרסה חזקה יותר של הבעיה (תנאי Validity חזק יותר). אם לפחות אחד מהקלטים הוא 0, אזי שני הקודקודים צריכים לפלוט 0.
הערה: נשים לב שהאלגוריתם ההסתברותי שהצענו מספק את תנאי זה. על מנת להוכיח חסם תחתון, נניח שאם שני הקלטים הם 1, אזי:

א.1 אם שום הודעה לא אבדה, שני הפלטים חייבים להיות 1.

א.2 אחרת, הקודקודים צריכים לפלוט אותו ערך עם הסתברות של לפחות $(1-\varepsilon)$.

משפט: במודל החזק יותר של שני הגנרלים, אם אחד הקודקודים צריך להחליט ב־r סיבובים, אזי ההסתברות לשגיאה הוא $\varepsilon \geq \frac{1}{r}$

הוכחה: תחילה נגדיר את הסימונים הבאים:

$$q_u := \Pr[\text{u outputs } 0]$$
$$q_v := \Pr[\text{v outputs } 0]$$

- ע"פ **א.1** בהרצה הראשונה בה הקלטים הם 1 ושום הודעה לא אבדה, הפלט חייב להיות 1.

- בהרצה השנייה E_1 כל ההודעות מגיעות חוץ מההודעה ש־v שולח ל־u. נשים לב שאם v הגריל אותו מספר, וההרצות דומות ההחלטה שלו תיהיה בדיוק אותו דבר. כיוון שההסתברות שהוא יוציא אותו מספר נשארת בדיוק אותו דבר כמו בהרצה הקודמת, כך גם ההסתברות ש־v פולט אפס, כלומר $q_v = 0$. יחד עם זאת u כן שם לב להבדל, ואולי הוא יכול לעשות משהו אחר, הוא לא חייב להסכים עם v. אומנם האלגוריתם התחייב שיש הסכמה בהסתברות לפחות $1 - \varepsilon$, כלומר ההסתברות שלא נסכים היא לכל היותר ε, דהיינו $q_u \leq \varepsilon$. דרך נוספת לראות את זה:

$$\begin{aligned}\varepsilon &\geq \Pr[\text{Disagreement}] \geq \Pr[\text{v outputs 1 and u outputs 0}] \\ &= \Pr[1_v] + \Pr[0_u] - \Pr[1_v \text{or} 0_u] \geq 1 + (q_u) - 1 = q_u\end{aligned}$$

- בהרצה E_2 שני ההודעות בסיבוב האחרון לא מגיעות, בפרט u לא יכול להבדיל בין ההרצות ולפי טיעון קודם $q_u \leq \varepsilon$, על פני כל בחירה מקרית הוא עושה אותו דבר, וכל בחירה שהובילה אותו לאפס גם הפעם מובילה אותו לאפס. נראה ש־$q_v \leq 2\varepsilon$ (בפרט מתקיים $q_v \leq \varepsilon$ אומנם כעת נסתפק בטענה חלשה יותר).

$$\begin{aligned}\varepsilon &\geq \Pr[\text{Disagreement}] \geq \Pr[\text{v outputs 0 and u outputs 1}] \\ &= \Pr[1_v] + \Pr[0_u] - \Pr[1_v \text{or} 0_u] \geq (1 - q_u) + (q_v) - 1 \geq q_v - \varepsilon \implies \boxed{q_v \leq 2\varepsilon}\end{aligned}$$

- E_{2r} שום הודעה לא התקבלה, ושני הקלטים הם 1. כאשר בה"כ $q_v \leq 2r\varepsilon$

- E_{2r+1} שום הודעה לא מגיע, אומנם כעת הקלט של v הוא אפס, עדיין מתקיים $q_u \leq 2r\varepsilon$ ו־$q_u = 1$ שכן ע"פ המודל החזק אם אחד הקלטים לפחות 0 אז בהכרח הפלט של שני הקודקודים היא 0, כלומר מוציאים אפס בהסתברות 1, לפיכך נקבל את אי השוויון:

$$2r\varepsilon \geq 1 \implies \boxed{\varepsilon \geq \frac{1}{2r}}$$

■

אלגוריתמי שידור ברשתות

3.1 Broadcast, Convergecast and Spanning Trees

3.1.1 העברת מסרים בטופולוגיות שירותיות

טופולוגית רשת נתונה באמצעות גרף מכוון $G := (V, E)$. בפרק זה נעסוק בבעיות עברת הודעות בצורה א-סינכרוניות ללא נפילות/כשלונות:

- הודעות תמיד נמסרות לאחר זמן סופי (מתזמן admissible).
- עיכוב הודעות הם בלתי ניתנות לחיזוי.

האלגוריתמים בהם נעסוק יהיו במובססות אירועים, כלומר *Event Base*.

- **שידור** (Broadcast): קודקוד v רוצה לשלוח הודעה לכל הקודקודים ברשת.
- **הצפה** (Flooding): כאשר קודקוד מקבל הודעה M בפעם הראשונה, הוא מעביר אותה לכל שכניו חוץ מלשכן ששלח לו את ההודעה.

3.1.2 הצפה במערכת סינכרונית

מערכת סינכרונית: הזמן מחולק לסיבובים מסונכרנים: כאשר הודעה מתעכבת בסיבוב אחד. **סיבוכיות זמן:** הסיבוכיות היא מספר הסיבובים.

הגדרות

- בהינתן קודקוד $V \in v$, הרדיוס של v בגרף G מוגדר $\mathbf{rad}_G(v) := \sup_{u\in V} \{d_G(u,v)\}$ כאשר $d_G(v)$ מוגדר כמרחק בין קודקוד u לקודקוד v בגרף G.
- הרדיוס של הגרף $\mathbf{rad}(G) := \min\{rad_G(v) : v \in V\}$
- הקוטר של הגרף $\operatorname{diam}(G) := \sup_{v\in V} \{\mathbf{rad}_G(v) : v \in V\} = \sup_{v,u\in V} \{d_G(v,u)\}$

כלומר הסיבוכיות זמן של אלגוריתם ההצפה הוא $\mathbf{rad}_G(v)$ **הבחנה:**

$$\operatorname{diam}(G)/2 \le \mathbf{rad}(G) \le \operatorname{diam}(G)$$

3.1.3 סיבוכיות לאן במערכות אסינכרוניות

תחילה נדגיש כי לא ברור כיצד נגדיר בכלל את סיבוכיות הזמן במערכת אסינכרונית, שכן היא לא פועלת לפי זמן, אלא לפי פעולות.
הנחות: עיכוב הודעה והזמן לחישובי צד (חישוביים מקומיים) הוא שרירותי. יחד עם זאת בעת ניתוח האלגוריתם נניח שעיכוב הודעה הוא לכל היותר יחידה זמן אחת (לא שנייה אחת). בנוסף נניח שחישובים מקומיים לוקחים אפס יחידות זמן. כעת לכל הרצה נמדוד את זמן הריצה ביחידות הזמן שהגדרנו. סיבוכיות הזמן תיהיה זמן הריצה של ההרצה עם הזמן המקסימלית (כלומר לכל הרצה זמן ההרצה קטן או שווה לסיבוכיות הזמן).
נשייך זמן לשליחה וקבלה של אירוע כך שמתקיים:

- סדר האירועים נשמר ללא שינוי
- חישוביים מקומיים לוקחים 0 יחידות זמן
- עיכוב הודעה לוקח לכל היותר יחידת זמן
- האירוע הראשון שנשלח, נשלח בזמן 0.
- זמן האירוע האחרון הוא ממוקסם.

הגדרה (סיבוכיות ריצה במערכת אסינכרונית): סך הזמן של ההרצה הכי גרועה בה חישובים לוקחים 0 זמן וכל ההודעות מתעכבות לכל היותר ביחידת זמן אחת.

3.1.4 סיבוכיות אלגוריתם ההצפה

משפט: סיבוכיות זמן הריצה של **הצפה** ממקור v ברשת אסינכרונית הוא $\mathbf{rad}_G(v)$

הוכחה: נראה שסיבוכיות הזמן $\geq \mathbf{rad}_G(v)$ וגם שסיבוכיות הזמן $\leq \mathbf{rad}_G(v)$.

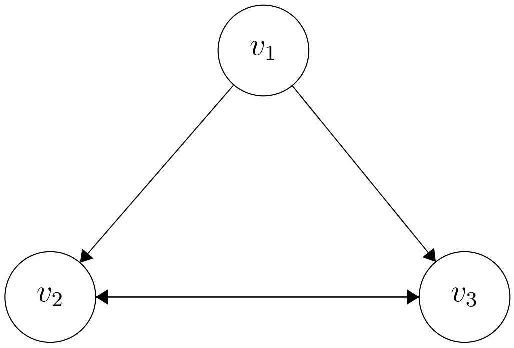
- יהי $u \in V$ כך ש-$d_G(u,v) = \mathbf{rad}_G(v)$. קיימת הרצה בה כל הודעה מתעכבת ביחידת זמן אחת, לפיכך סיבוכיות הזמן $\geq \mathbf{rad}_G(v)$.
- יהי $u \in V$ קודקוד שרירותי ויהי $t = d_G(u,v)$, בברור מתקיים $0 \leq t \leq \mathbf{rad}_G(v)$. נוכיח באינדוקציה על t שהזמן שלוקח להודעה להגיע ל-u הוא לכל היותר t.
בסיס: עבור $t=0$ הטענה נכונה.
צעד: נניח שהטענה נכונה עבור t ונוכיח עבור $t+1$. יהי $v, w_1, w_2, ..., w_t, u$ מסלול באורך $t+1$ מ-v ל-u. ע"פ הנחת האינדוקציה, w_t מקבל את ההודעה מ-v בזמן $\leq t$. אם w_t מעביר את ההודעה ל-u, זה לוקח זמן ≤ 1. סה"כ u מקבל את ההודעה עד זמן $t+1$ לכל היותר. אם v לא קיבל את ההודעה מ-w_t אזי הוא v קיבל את ההודעה לפני w_t ומהנחת האינדוקציה הטענה נובעת. ∎

הגדרה (סיבוכיות ההודעה): מספר ההודעות שנשלחו סה"כ.

משפט: סיבוכיות ההודעה של **הצפה** בגרף $G = (V, E)$ היא $O(|E|)$

הוכחה: יהי $uw \in E$ צלע שרירותית בגרף G. ע"פ האלגוריתם u שולח לכל היותר הודעה אחת ל-w ו-w שולח לכל היותר הודעה אחת ל-u. לכן מספר ההודעות סה"כ הוא $2 \cdot |E|$ לכל היותר. ■

דוגמה למקרה בה בקשת עוברים שתי הודעות:



3.1.5 Flooding Spanning Tree

קודקוד המקור v הוא השורש של העץ.
עבור כל שאר הקודקודים, השכן שהעביר להם את ההודעה הוא ההורה (אם יש כמה בחר אחד שרירותית).

3.1.6 עץ פורש במערכות סינכרוניות

במערכות סינכרוניות, קודקוד u ישיג בסיבוב t אם ורק אם $d_G(u,v) = t$, לפיכך המרחק של קודקוד u מהשורש שווה לסיבוב בו u הפך לישיג (מ-v), לכן העץ הפורש משמר את המרחקים מהשורש בגרף G, עצים כנ"ל נקראים עצי
Shortest Path או Breadth First Search

משפט: במערכת אסינכרונית, אלגוריתם ההצפה בונה עץ BFS.

3.1.7 עץ פורש במערכות אסינכרוניות

הבחנה: במערכות אסינכרוניות, כל עץ פורש יכול להבנות באמצעות אלגוריתם ההצפה, בפרט העומק של העץ יכול להיות גדול כ-$n-1$ אפילו אם הרדיוס/הקוטר של הגרף הוא 1.

3.1.8 Converge-cast

זהו ההפך משידור (כלומר Broadcast): בהינתן שורש של עץ פורש, כל הקודקודים שולחים הודעה לשורש.

האלגוריתם:

- **איתחול עבור עלה** $u \in V$ שלח את ההודעה להורה.
- **איתחול עבור קודקוד פנימי** w: עד לקבלת הודעה מילד x: אם w קיבל הודעות מכל ילדיו, שלח הודעה להורה.
- **עבור קודקוד השורש** v: עד לקבלת הודעה מהילד x: אם v קיבל הודעה מכל הילדים, אזי האלגוריתם מסתיים.

Convergecast algorithm: Analysis & Remarks

- סיבוכיות הזמן: עומק העץ
- סיבוכיות ההודעה: מספר הצלעות של העץ(כלומר $|V|-1$).
- אפליקציות: חישוב מינימום, מקסימום, סכום, ממוצע וכיו"ב.

3.2 אלגוריתם ההדהוד/ההצפה

בהינתן שורש, הצפה ו-converge-cast ניתנים לשימוש באופן הבא:

1. השתמש באלגוריתם ההצפה לבנות עץ.
2. השתמש ב-converge-cast(echo)(הדהוד) להודיע לשורש על הסיום.

זמן הריצה: עומק העץ

- במערכות סינכרוניות $O(\text{diam}\,(G))$ כי בנינו עץ BFS
- במערכות אסינכרוניות $O(|V(G)|)$ כי היריב בנו לנו עץ שהעומק שלו הוא $|V(G)|$, והסיבוכיות היא רדיוס כלפי מטה, ועומק העץ כלפי מעלה.

במערכות אסינכרוניות עומק של העץ הנבנה יכול להיות $\Omega(|V(G)|)$, אפילו אם הקוטר של G הוא 1. איך ניתן לבנות עץ עם עומק קטן יותר?

3.3 בניית עץ מסלולים מינימלי

3.3.1 אלגוריתם דיקסטרה

בשלב t: תת–העץ של כל הקודקודים שמרחקם הוא לכל היותר r_t מקודקוד המקור v.
בצעד הבא: הוסף לעץ קודקוד שהמרחק שלו מקודקוד v הוא מינימלי.

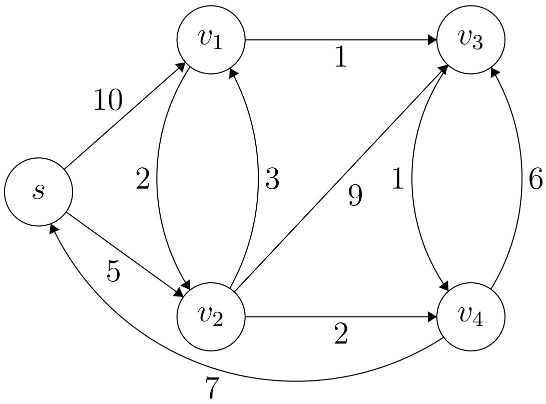
3.3.2 אלגוריתם בלמן–פורד

עבור כל קודקוד u שמר מרחק מוערך d_u לקודקוד המקור v.

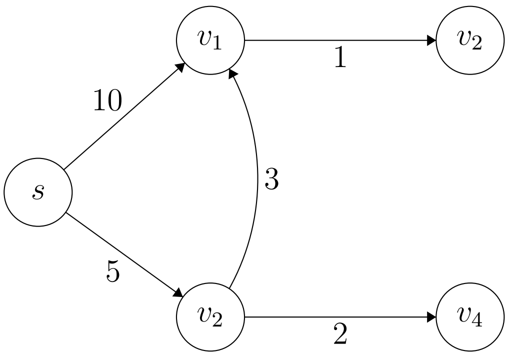
- **אתחל:** $d_v = 0$ וגם $d_u = \infty$ לכל קודקוד $u \neq v$.
- בכל שלב, כל הקודקודים מעדכנים את המרחק המשוערך שלהם בהתבסס על הערך המשוערך של שכניהם:

$$d_v = \min\left\{d_v, \min_{u\in N(v)}\{d_u+1\}\right\}$$

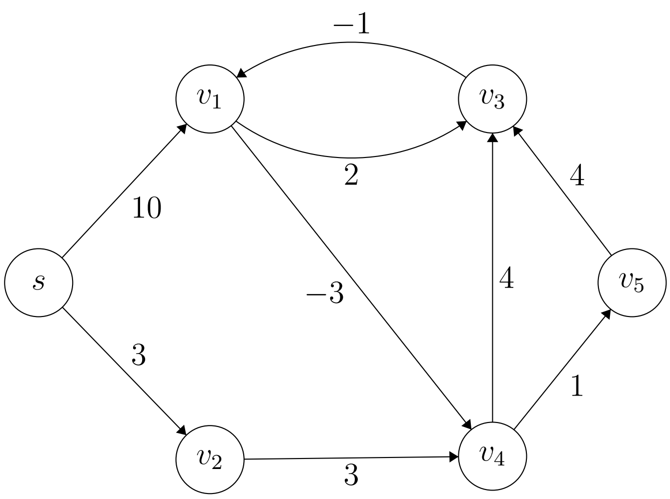
תרגיל: הרץ את האלגוריתמים על הגרף הבא:



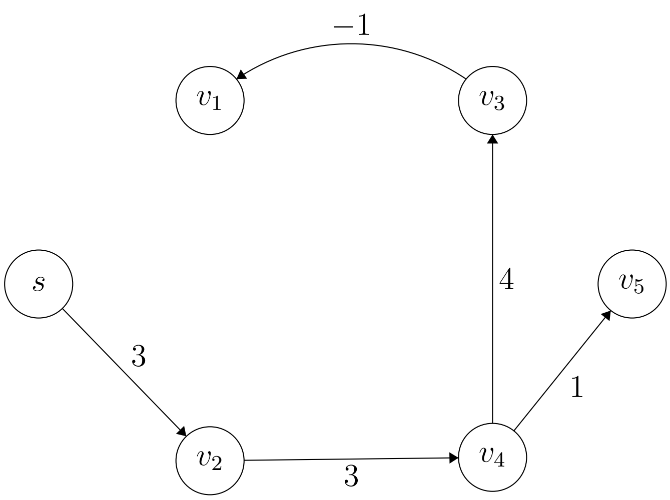
פתרון: $d_s = 0,\ d_{v_1} = 8,\ d_{v_2} = 5,\ d_{v_3} = 13,\ d_{v_4} = 7$ תוצאת ההרצה על הדוגמה:



הרץ את BF על הגרף הבא:



דוגמה לעץ שמתקבל:



3.3.3 אלגוריתם דיקסטרה המבוזר

במקרה שלנו הגרף לא ממשוקל, לכן נוכל לגדל את העץ רמה רמה, כמו במערכת סינכרונית.
נניח שבנינו עץ עד למרחק r מקודקוד המקור v, כיצד נוסיף את הרמה הבאה?
קודקוד המקור יתאם את שלבי האלגוריתם.
שלב $r+1$ של האלגוריתם:

1. קודקוד המקור משדר "התחל שלב $r+1$" בעץ הנוכחי.
2. העלים של העץ הנוכחי (העלים בעומק r) שולחים "הצטרף לשלב $r+1$" לכל השכנים **החדשים** שלהם.
3. קודקוד u מקבל את ההודעה "הצטרף לשלב ה-$r+1$ מקודקוד w:

- אם זאת ההודעה הראשונה ש־u מקבל הוא מגדיר $p(u)=w$ ומחזיר ל-w את ההודעה ACK.
- אחרת החזר $NACK$ ל-w.

סיבוכיות זמן ריצה: עבור כל שכבה מבצעים $O(r)$ flooding לעלים, כיוון ששולחים הודעה לכל השכנים והם מחזרים וכל הודעה מעוקבת באחד לכל היותר, ולבסוף מחזירים את ההודעה לשורש שזה $O(r)$. סה"כ $O(r+2+r)=O(r)$ להוספה של רמה. סה"כ $O(\mathrm{diam}\,(G)^2)$

3.3.4 אלגוריתם בלמן-פורד המבוזר

3.3.5 אלגוריתם בניית עץ ה-BFS המבוזר: סיכום

סיבתיות, זמן לוגי, ומצבים גלובלים

4.1 שעונים לוגיים

מטרה: לתת לכל אירוע חותמת זמן במערכת העברת הודעות אסינכרוניות.

- מאפשר לנו לתת לקודקודים סימון כלשהו לזמן.
  - האלגוריתמים יכולו להשתמש בזמן.
- **ערכי שעון לוגי:** ערכיים מספריים שגדלים במהלך הזמן וקונסיסטנטים (עקביים) עם מה שקרה במערכת, ליתר דיוק, עם פעולות ברי אבחנה ע"י המערכת.
- השעון הלוגי הוא **לא** בשביל לבצע **סינכרון שעונים**.
  סינכרון שעונים: חישוב שעונים לוגים בכל הקודקודים שמסמלץ זמן אמיתי, ומסונכן באופן הדוק.

4.2 התנהגויות ברי אבחנה

תזכורת: הרצות ומתזמנים

- הרצה היא סדרה מתחלפת של קונפיגורציות ואירועים.
- מתזמן $\mathcal{S}$ הוא סדרת האירועים של הרצה.
  - היתכן שיכלול קלטי קודקודים.
- Schedule restriction לקודקוד v, מסומן באמצעות $\mathcal{S}|v$, הוא סדרת האירועים ב–$\mathcal{S}$ שנראו ע"י v.

Causal Shuffles

הגדרה (Causal shuffle): מתזמן $\mathcal{S}'$ הוא causal shuffle (מערבב סיבתי) של מתזמן $\mathcal{S}$ אם $\mathcal{S}'|v = \mathcal{S}|v$ לכל $v \in V$.

אבחנה: המערכת המבוזרת לא יכולה להבחין בין $\mathcal{S}$ ו-$\mathcal{S}'$ אם ורק אם $\mathcal{S}'$ הוא causal shuffle של $\mathcal{S}$

Causal Order

הגדרה (מופע מוכח/קרה באופן מוכח): במתזמן $\mathcal{S}$ אירוע e יקרא מופע מוכח לפני אירוע e' אם e מופיע לפני e' בכל מערבב סיבתי (causal shuffle) של $\mathcal{S}'$

שעונים לוגים מבוססים על סדר סיבתי (causal order) של אירועים.

- אם אירוע e קרה באופן מוכח לפני אירוע e', אזי e אמור להופיע לפני e' בסדר הסיבתי (causal order).
- במקרה זה, ערך השעון של e אמור להיות קטן מערך השעון של e'.

4.3 Lamport′s Happens – Before Relation

הנחה: מערכת העברת הודעות, רק שולחת ומקבלת אירועים.

- נשקול שני אירועים e ו–e' במתזמן $\mathcal{S}$ שמופעים בקודקודים u ו–u', בהתאמה.
  - אירוע שליחת ההודעה מופיע בקודקוד השולח, ואירוע קבלת ההודעה מופיע בקודקוד המקבל.
  - e מופיע בזמן t ו–e' מופיע בזמן t'.

בכל אחד מהמקרים הבאים אנו יודעים ש–e מופיע באופן מוכח לפני e'.

1. $u = u'$ וגם $t < t'$.
2. e' הוא אירוע קבלה שמתאים לאירוע השולח e.
3. קיים אירוע e'' שעבורו אנו יודעים ש-e מופיע באופן מוכח לפני e'' ו-e'' מופיע באופן מוכח לפני e'.

הגדרה (יחס קורה-לפני): היחס קורה-לפניבהתחשב במתזמן $\mathcal{S}$ הוא יחס בינארי על אירועי שליחה וקבלה ב-$\mathcal{S}$ היחס יסומן ב- $\Rightarrow_{\mathcal{S}}$. היחס מכיל:

א. כל הזוגות (e, e') כך ש-e קדם ל-e' ב-$\mathcal{S}$ ולשניהם מופיעים באותו קודקוד.

ב. כל הזוגות (e, e') כך ש-e הוא אירוע שליח ו-e' הוא אירוע קבלה של אותה הודעה.

ג. כל הזוגות (e, e') עבורם קיים אירוע שלישי e'' כך ש-e'' $\Longrightarrow_{\mathcal{S}}$ e וגם e' $\Longrightarrow_{\mathcal{S}}$ e''.

אבחנה: היחס $\Longrightarrow_{\mathcal{S}}$ סגור תחת טרנזטיביות, שמוגדר ע"י א, ב.

4.4 Happens – Before and Causal Shuffles

משפט: עבור מתזמן $\mathcal{S}$ ושני אירועים (שליחה/קבלה) e ו-e', שני המשפטים הבאים שקולים:

א. אירוע e קורה לפני אירוע e', כלומר $e \Longrightarrow_{\mathcal{S}} e'$

ב. אירוע e קדם לאירוע e' בכל הערבובים הסיבתיים (in all causal shuffles) $\mathcal{S}'$ של $\mathcal{S}$

הערה: המשפט הנ"ל מראה שיחס הקורה לפני תופס בדיוק את הסיבתיות בין האירועים, כלומר זה תופס את הכל בקשר לאירועים שניתנים להבחנה ע"י המערכת.

הוכחה: נראה ש-(2) $\implies$ (1) וגם (1) $\implies$ (2)

- **1** $\implies$ **2**, כלומר נניח ש-$e' \Leftrightarrow_{\mathcal{S}} e$, אזי e קודם ל-e' בכל ה-causal shuffle$\mathcal{S}'$ של $\mathcal{S}$.
  ההוכחה היא באינדוקציה על k–מספר האירועים ב-$\mathcal{S}$ בין e ל-e'.
  בסיס: עבור $k = 0$ e ו־-e' מופיעים באותו קודקוד או שהם הודעות שליחה וקבלה של אותה הודעה, ולכן הטענה תקפה.
  צעד האינדוקציה: נניח שהטענה נכונה כל $n \leq k$ ונראה שהטענה נכונה עבור $k+1$.אם e וגם e' מופיעים באותו קודקוד או אם הם אירועי שליחה וקבלה של אותה הודעה, הטענה תקפה. אחרת, קיים אירוע e'' כך ש-$e \Longrightarrow_{\mathcal{S}} e''$ ו-$e'' \Longrightarrow_{\mathcal{S}} e'$. בברור מספר האירועים ב-$\mathcal{S}$ בין e ל-e'' ובין e'' ל-e' הוא לכל היותר k. לפיכך, ע"פ הנחת האינדוקציה, e קודם ל-e'' וגם e'' קודם ל-e' בכל causal shuffle של $\mathcal{S}'$ של $\mathcal{S}$. בפרט נסיק ש-e קודם ל-e' בכל causal shuffle $\mathcal{S}'$ של $\mathcal{S}$ כפי שנטען.

- **(1)** $\implies$ **(2)**: אם e קודם ל-e' בכל causal shuffle $\mathcal{S}'$ של $\mathcal{S}$, אזי $e \Longrightarrow_{\mathcal{S}} e'$. ■

קונצנזוס

5.1 ליכרון מסוף

5.1.1 הגדרות

- הרצה בה אחד מהקודקודים מחליט על הפלט ללא תלות בשאר הקודקודים תיקרא הרצה יחידנית.
- קודקוד יקרא $valent - x$ אם כל המשך ממנו יוביל לפלט x, כלומר כל העלים שלו פולטים x.
- קודקוד יקרא BIVALENT אם ניתן להגיע ממנו לפלט 1 וגם ל-0, כלומר יש ענף ממנו שיפלוט 0, וענף שיפלוט 1. נאמר שקודקוד הוא קריטי אם אחד הבנים שלו הוא $valent - 0$ והשני $valent - 1$. **אבחנה:** אם קודקוד הוא *bivalent* ולא קריטי אזי אחד מילדיו הוא *bivalent*, כלומר קודקוד קריטי הוא קודקוד ה-*bivalent* האחרון (באופן שתואר).

  אבחנה נוספת: אם קודקוד *bivalent* אזי כל הורה שלו *bivalent*.

משפט: קיים קלט עבורו המצב ההתחלתי הוא ביוולנטי.

הוכחה: נניח ששני הקלטים הם 0, ע"פ תקפות *validity* הפלטים יהיו 0. ■

5.2 אלגוריתם המלכה

5.2.1 הוכחת נכונות

נוכיח שתי טענות

משפט: אחרי השלב שבו המלכה נכונה, כל הקודקוד הנכונים בעלי אותו ערך

הוכחה: ישנם שתי אפשרויות:

- אם כל הקודקודים שינו את ערכם לערך של המלכה, לכולם יש את אותו ערך.
- נניח שבשלב המדובר קודקוד תקין x לא מקשיב למלכה, כלומר תומך בערך שלו, לכן בסיבוב הראשון של השלב הזה x קיבל את הערך (נניח a) יותר מ-$\frac{n}{2}$ פעמים. כיוון שיש רק f קודקודים ביזנטיים יותר מ-$\frac{n}{2}$ קודקודים תקינים שלחו ל-x את הערך a.

למעשה כל קודקוד תקין (בפרט המלכה) קיבלו את הערך יותר מ-$\frac{n}{2}$ פעמים. כיוון שלא יתכן ששני ערכים שונים התקבלו יותר מ-$\frac{n}{2}$ פעמים, בסוף הסיבוב הראשון הערך של כל הקודקודים התקינים הוא a – כולל המלכה. כעת כל קודקוד תקין מקשיב למלכה וערכו a או שאינו תומך במלכה וערכו הוא a, בשני המקרים הערך של כל קודקוד תקין הוא a. ■

משפט: בכל השלבים הבאים, אף קודקוד לא משנה את ערכו.

הוכחה: נניח שבתחילת שלב מסוים הערך של כל הקודקודים התקינים הוא a, כל קודקוד תקין יקבל בסיבוב הראשון את הערך a לפחות $n-f$ פעמים. אם $n-f > \frac{n}{2}+f$, כל התקינים יתמכו ב-a ולא יקשיבו למלכה. בסוף השלב ערכם עדיין a. כעת נשים לב שמתקיים

$$n-f > \frac{n}{2}+f \iff \frac{n}{2} > 2f \iff \boxed{\frac{n}{4} > f}$$

■

5.3 אלגוריתם המלך

אלגוריתם המלך מסוגל להתמודד עם $f < \frac{n}{2}$ כשלונות ביזנטים, ומשתמש בהודעות קטנות. **הרעיון העיקרי** הרעיון דומה לרעיון של אלגוריתם המלכה, קיים מלך שונה (הידוע מראש) בכל שלב. כיוון שיש $f+1$ סיבובים, קיים סיבוב בו המלך אינו ביזנטי. **ההבדל העיקרי** הוא שקיים שלב ביניים בו הקודקודים מצעים ערך אם הם קיבלו אותו הרבה פעמים. ערך מוצע יתקבל אם קודקודים רבים מצעים אותו.

5.3.1 נכונות

אבחנה: אם קודקוד נכון (לא ביזנטי) הציע x, אז שום קודקוד נכון אחר הציע $y \neq x$.

הוכחה: נניח בשלילה שקיימים שני קודקודים נכונים שהציעו שני ערכים שונים, כיוון שכל קודקוד קיבל את הערך לפחות $n-f$ ולכל היותר יש f קודקודים ביזנטים אז כל אחד קיבל לפחות $n-2f$ ערכים אמיתיים, מקודקודים אמיתיים זרים. סה"כ מספר הקודקודים הוא לפחות $2(n-2f)+f = 2n-3f > n$ קודקודים, סתירה. ■

משפט: לאחר השלב בו המלך הוא אמיתי (לא ביזנטי), כל הקודקודים בעלי אותו ערך.

הוכחה: נשקול את המקרים הבאים:

- אם כל הקודקודים שינו את ערכם לערך של המלך, אז ברור שכל הערכים זהים.
- אחרת, קיים קודקוד כלשהו שלא שינה את ערכו לערך שלל המלך, אבל מקרה זה יתרחש רק אם הוא קיבל הצעה לפחות $n-f$ פעמים. כלומר, לפחות $n-2f$ קודקודים נכונים שלחו את ההצעה הזאת, וכל הקודקודים האמיתיים קיבלו אותו לפחות $n-2f > f$ פעמים, לפיכך כל הקודקודים שינו את ערכם לערך המוצע. שים לב שרק ערך אחד יכול להיות מוצע יותר מ-f פעמים. ■

משפט: בכל השלבים הבאים אף קודקוד נכון לא משנה את ערכו.

הוכחה: זה נובע ישירות מהעובדה שכל הקודקודים האמיתיים בעלי אותו ערך אחרי השלב בו המלך הוא אמיתי, ומהתקפות ($validity$). ■