<u>הגדרה:</u>
PSPACE = {L | קיימת מ"ט דטרמיניסטית המקבלת את L בזיכרון פולינומי}

<u>משפט:</u> $P \subseteq PSPACE$
הוכחה: בזמן פולינומי ניתן לברק במספר פולינומי של תאים, ולכן כל מ"ט שעובדת בזמן פולינומי גם בזיכרון פולינומי.

<u>משפט:</u> $PSPACE \subseteq R$
הוכחה: תהא $L \in PSPACE$ ותהה M מכונה המקבלת (לאו דווקא מכריעה) אותה בזיכרון פולינומי, $p(n)$, אך לא בהכרח עוצרת לכל קלט.

<u>אבחנה 1:</u> הראש יכול להיות לכל היותר ב $p(n)$ מקומות. מספר המצבים הוא $|Q|$ ויש $|\Gamma|^{p(n)}$ אפשרויות לתוכן הסרט. סה"כ מספר הקונפיגורציות האפשריות הוא: $D(n) = p(n) \cdot |Q| \cdot |\Gamma|^{p(n)}$.

<u>אבחנה 2:</u> אם M רצה יותר מ $D(n)$ צעדים אז היא לא עוצרת לעולם (ובפרט לא מקבלת).

נבנה מכונה M' שמכריעה את L:
M' על קלט x:
1. מחשבת את $D(|x|)$.
2. מריצה את M על x תוך כדי ספירת הצעדים של M.
3. אם M קיבלה אז M' מקבלת.
4. אחרת, אם מספר הצעדים גדול מ $D(|x|)$ אז עוצרת ודוחה.

<u>נכונות:</u> נובע מאבחנה 2.
<u>פולינומיות:</u>
כתיבת מספר לוקחת log של המספר תאים.
לכן כתיבת $D(|x|)$ לוקחת $\log(D(|x|)) = \log(p(|x|)) + \log(|Q|) + p(|x|) \cdot \log(|\Gamma|)$
לכן החישוב של $D(|x|)$ דורשת מס' תאים פולינומי.
סה"כ ההרצה דורשת מספר תאים פולינומי: $O(\log(D(|x|))) + p(|x|) = O(p(|x|))$

מסקנה: אם $L \in PSPACE$ אז קיימת מכונה <u>שמכריעה</u> את L בסיבוכיות מקום פולינומית.
לכן אפשר להגדיר את $PSPACE$ כמחלקת השפות שקיימת עבורן מ"ט המשתמשת בזיכרון פולינומי ועוצרת לכל קלט.

<u>דוגמה 1:</u>
$PRIMES = \{x \mid$ ראשוני הנתון בייצוג בינארי $x\}$
נשים לב שסיבוכיות הזיכרון צריכה להיות פולינומית באורך הקלט, כלומר $\log(x)$.
האם $PRIMES \in PSPACE$? כן!
נתאר מ"ט M_{PRIMES} המכריעה את השפה תוך שימוש בזיכרון פולינומי.
M_{PRIMES} על קלט x עוברת על כל המספרים הקטנים מ x ועבור כל מספר y כזה, בודקת האם $x \bmod y = 0$ ואם קיים y כזה אז דוחה. אם לכל y מתקיים $x \bmod y \neq 0$ אז מקבלת.
<u>נכונות:</u> המכונה מקבלת את x אמ"מ הוא ראשוני.

<u>סיבוכיות זיכרון:</u>
גודל הייצוג של כל מספר קטן מ x הוא $O(\log(x))$.
המכונה זקוקה למספר אחד כזה, היא מוסיפה לו כל פעם אחד ובנוסף זקוקה למספר דומה של תאים לביצוע החלוקה.
סה"כ $O(\log(x))$ ולכן ליניארי בגודל הקלט ולכן פולינומי בגודל הקלט.

נשים לב שהמכונה שתיארנו עובדת בזמן $O(x)$ ולכן סיבוכיות הזמן אקספוננציאלית בגודל הקלט.
(זהו כמובן לא אומר שאין מ"ט אחרת שמקבלת / מכריעה את השפה בזמן פולינומי ואפילו בשנת 2002 פורסם אלגוריתם שעושה זאת).

<u>דוגמה 2:</u>
ראינו $P \subseteq PSPACE \subseteq R$ אבל לא ידוע האם $P = PSPACE$ או לא. כן ידוע $PSPACE \subset R$.
בעיית הקליק: (ידוע שהיא ב $PSPACE$. לא ידוע האם היא ב P)
הקלט: גרף לא מכוון $G = (V, E)$ ומספר טבעי $k \le |V|$.
(קליק הוא תת קבוצה של צמתים שבין כל זוג צמתים בתת הקבוצה מחברת קשת).
הבעיה: האם ב G קיים קליק עם לפחות k צמתים.
$L = \{(\langle G\rangle, k) \mid$ ב G יש קליק בגודל $k\}$ כאשר $\langle G\rangle$ הוא קידוד של גרף, למשל ע"י מטריצת שכנויות.
M_L על $(\langle G\rangle, k)$ עוברת על כל תתי הקבוצות (בגודל k). עבור כל תת קבוצה בודקת את הקשתות.
אם נמצא קליק בגודל k מקבלת. אחרת, אם עברה על כל תתי הקבוצות ולא מצאה, אז דוחה.
<u>נכונות</u>: המכונה מקבלת אמ"מ יש קליק בגודל k בגרף G.
<u>סיבוכיות זיכרון:</u>
כל ווקטור בינארי באורך $|V|$ מייצג תת קבוצה.
המכונה זקוקה לווקטור אחד כזה ובנוסף למספר קבוע של תאים לבדיקה.
לכן הזיכרון הדרוש הוא ליניארי בגודל הייצוג ולכן פולינומי בגודל הקלט.
<u>סיבוכיות הזמן:</u> אקספוננציאלית אבל יתכן שקיימת מ"ט המקבלת את השפה בזמן פולינומי.

הערה: לכל k קבוע השפה ב P.
מספר תתי הקבוצות האפשרי הוא:

$$\binom{|V|}{K} = \frac{|V|(|V|-1)\cdot\ldots}{(|V|-k)!k!} = \frac{|V|(|V|-1)\cdot\ldots\cdot(|V|-k+1)}{k!} = O\left(|V|^k\right)$$

נשים לב שבשפה שלנו k הוא חלק מהקלט ואינו קבוע.