吴 铤

2017年12月24日



1 萨维奇定理

2 PSPACE类

3 PSPACE完全性

1 萨维奇定理

2 PSPACE类

3 PSPACE完全性

1 萨维奇定理

2 PSPACE类

3 PSPACE完全性

定义1 (空间复杂度)

令M是在所有輸入上都停机的确定型图灵机,M的空间复杂度是函数 $f: \mathbb{N} \to \mathbb{N}$,其中f(n)是M在任何长为n的输入上扫描带方格的最大数。

若M的空间复杂度为f(n),则称M在空间f(n)内运行。

如果M是在所有输入上都停机的非确定型图灵机,则 其空间复杂度为M在任何长为n的输入上,在任何计算分支 上所扫描的带方格的最大数。

定义2 (空间复杂度类)

令 $f: \mathbb{N} \to \mathbb{R}^+$ 是一个函数,则定义

- $SPACE(f(n)) = \{L : L = L = L \}$ L是被O(f(n))空间的确定性图灵机判定的语言 $\{L = L \}$

定义2 (空间复杂度类)

令 $f: \mathbb{N} \to \mathbb{R}^+$ 是一个函数,则定义

- $SPACE(f(n)) = \{L :$ L是被O(f(n))空间的确定性图灵机判定的语言 $\}$

定义2(空间复杂度类)

令 $f: \mathbb{N} \to \mathbb{R}^+$ 是一个函数,则定义

- $SPACE(f(n)) = \{L :$ L是被O(f(n))空间的确定性图灵机判定的语言 $\}$

■ 对于输入的⟨φ⟩, φ是布尔公式

 \blacksquare 对于 ϕ 中的变量 x_1, \dots, x_m 的每个赋值

☑ 若φ的值为1,则接受;否则拒绝。

- 对于输入的⟨ø⟩,ø是布尔公式
 - 1 对于 ϕ 中的变量 x_1, \dots, x_m 的每个赋值
 - 计算φ在该真值下的值
 - 2 若 ϕ 的值为1,则接受;否则拒绝。



- 对于输入的⟨φ⟩,φ是布尔公式
 - 11 对于 ϕ 中的变量 x_1, \dots, x_m 的每个赋值
 - 计算φ在该真值下的值
 - 2 $\Xi \phi$ 的值为1,则接受;否则拒绝。



- 对于输入的⟨ø⟩,ø是布尔公式
 - \blacksquare 对于 ϕ 中的变量 x_1, \cdots, x_m 的每个赋值
 - 计算φ在该真值下的值
 - 2 若 ϕ 的值为1,则接受;否则拒绝。



- 对于输入的⟨ø⟩,ø是布尔公式
 - 11 对于 ϕ 中的变量 x_1, \dots, x_m 的每个赋值
 - 计算φ在该真值下的值
 - 2 若 ϕ 的值为1,则接受;否则拒绝。



对于任意满足
$$f(n) \ge n$$
的函数 $f: \mathbb{N} \to \mathbb{R}^+$,有

$$NSPACE(f(n)) \subseteq SPACE(f^2(n)).$$

- 可产生性问题子程序CANYIELD:
 - 输入NTM N在输入w上的格局 c_1, c_2 以及整数t
 - 判定N能合在t步内从 c_1 变换到 c_2

对于任意满足 $f(n) \ge n$ 的函数 $f: \mathbb{N} \to \mathbb{R}^+$,有 $NSPACE(f(n)) \subseteq SPACE(f^2(n))$.

- 可产生性问题子程序CANYIELD:
 - 输入NTM N在输入w上的格局 c_1, c_2 以及整数t
 - 判定N能否在t步内从 c_1 变换到 c_2

对于任意满足 $f(n) \ge n$ 的函数 $f: \mathbb{N} \to \mathbb{R}^+$,有 $NSPACE(f(n)) \subseteq SPACE(f^2(n))$.

- 可产生性问题子程序CANYIELD:
 - 输入NTM N在输入w上的格局 c_1, c_2 以及整数t
 - 判定N能合在t步内从 c_1 变换到 c_2

对于任意满足 $f(n) \ge n$ 的函数 $f: \mathbb{N} \to \mathbb{R}^+$,有 $NSPACE(f(n)) \subseteq SPACE(f^2(n)).$

- 可产生性问题子程序CANYIELD:
 - 输入NTM N在输入w上的格局 c_1, c_2 以及整数t
 - 判定N能否在t步内从 c_1 变换到 c_2

- - 检查c1 = c
 - 或根据N的规则检查 c_1 能否在一步内产生 c_2

如果两者有一个成立,则接受,否则拒绝。

- ② $\overline{A}t > 1$,对于每个格局 c_m 执行
 - 运行 $CANYIELD(c_1, c_m, t/2)$
 - **■** 运行 $CANYIELD(c_m, c_2, t/2)$

如果都接受,则接受

- 1 若t=1,
 - 检查 $c_1 = c_2$
 - 或根据N的规则检查 c_1 能否在一步内产生 c_2

如果两者有一个成立,则接受,否则拒绝。

- \mathbf{Z} 若t > 1,对于每个格局 c_m 执行
 - 运行 $CANYIELD(c_1, c_m, t/2)$
 - 运行 $CANYIELD(c_m, c_2, t/2)$

如果都接受,则接受



- 1 若t=1,
 - 检查 $c_1 = c_2$
 - \blacksquare 或根据N的规则检查 c_1 能否在一步内产生 c_2 如果两者有一个成立,则接受,否则拒绝。
- ② 若t > 1,对于每个格局 c_m 执行
 - 运行 $CANYIELD(c_1, c_m, t/2)$
 - 运行 $CANYIELD(c_m, c_2, t/2)$

如果都接受,则接受



- II 若t=1,
 - 检查 $c_1 = c_2$
 - 或根据N的规则检查 c_1 能否在一步内产生 c_2

如果两者有一个成立,则接受,否则拒绝。

- ② $\overline{A}t > 1$,对于每个格局 c_m 执行
 - 运行 $CANYIELD(c_1, c_m, t/2)$
 - 运行 $CANYIELD(c_m, c_2, t/2)$
 - 如果都接受,则接受
- 3 若还没有接受,则拒绝。

- 1 若t = 1,
 - 检查 $c_1 = c_2$
 - 或根据N的规则检查 c_1 能否在一步内产生 c_2

如果两者有一个成立,则接受,否则拒绝。

- ② 若t > 1,对于每个格局 c_m 执行
 - 运行 $CANYIELD(c_1, c_m, t/2)$
 - 运行 $CANYIELD(c_m, c_2, t/2)$
 - 如果都接受,则接受
- 若还没有接受,则拒绝。

- II 若t=1,
 - 检查 $c_1 = c_2$
 - 或根据N的规则检查 c_1 能否在一步内产生 c_2

如果两者有一个成立,则接受,否则拒绝。

- 2 若t > 1,对于每个格局 c_m 执行
 - 运行 $CANYIELD(c_1, c_m, t/2)$
 - 运行 $CANYIELD(c_m, c_2, t/2)$

如果都接受,则接受



- II 若t=1,
 - 检查 $c_1 = c_2$
 - 或根据N的规则检查 c_1 能否在一步内产生 c_2

如果两者有一个成立,则接受,否则拒绝。

- 2 若t > 1,对于每个格局 c_m 执行
 - 运行 $CANYIELD(c_1, c_m, t/2)$
 - 运行 $CANYIELD(c_m, c_2, t/2)$

如果都接受,则接受

- II 若t=1,
 - 检查 $c_1 = c_2$
 - 或根据N的规则检查 c_1 能否在一步内产生 c_2

如果两者有一个成立,则接受,否则拒绝。

- 2 若t > 1,对于每个格局 c_m 执行
 - 运行 $CANYIELD(c_1, c_m, t/2)$
 - 运行 $CANYIELD(c_m, c_2, t/2)$

如果都接受,则接受



- II 若t=1,
 - 检查 $c_1 = c_2$
 - 或根据N的规则检查 c_1 能否在一步内产生 c_2

如果两者有一个成立,则接受,否则拒绝。

- 2 若t > 1,对于每个格局 c_m 执行
 - 运行 $CANYIELD(c_1, c_m, t/2)$
 - 运行 $CANYIELD(c_m, c_2, t/2)$

如果都接受,则接受



- \blacksquare 确定型图灵机M对于输入的w
 - 输出CANYIELD $(c_{start}, c_{accept}, 2^{a_j(n)})$
- 递归的每一层增加的空间: O(f(n))
- 递归次数: $O(\log 2^{df(n)}) = O(f(n))$



- 确定型图灵机M对于输入的w
 - 输出CANYIELD($c_{start}, c_{accept}, 2^{df(n)}$)
- 递归的每一层增加的空间: O(f(n))
- 递归次数: $O(\log 2^{d_f(n)}) = O(f(n))$

对于NTM N,确定使得格局数不超过2^{df(n)}的常数d。

- 确定型图灵机M对于输入的w
 - 输出CANYIELD $(c_{start}, c_{accept}, 2^{df(n)})$

- 确定型图灵机M对于输入的w
 - 输出CANYIELD $(c_{start}, c_{accept}, 2^{df(n)})$
- 递归的每一层增加的空间: O(f(n))
- 递归次数:

- 确定型图灵机M对于输入的w
 - 输出CANYIELD $(c_{start}, c_{accept}, 2^{df(n)})$
- 递归的每一层增加的空间: O(f(n))
- 逓归次数:

- 确定型图灵机M对于输入的w
 - 输出CANYIELD $(c_{start}, c_{accept}, 2^{df(n)})$
- 递归的每一层增加的空间: O(f(n))
- 递归次数: $O(\log 2^{df(n)}) = O(f(n))$

- 确定型图灵机M对于输入的w
 - 输出CANYIELD $(c_{start}, c_{accept}, 2^{df(n)})$
- 递归的每一层增加的空间: O(f(n))
- 递归次数: $O(\log 2^{df(n)}) = O(f(n))$

定义3

- PSPACE: 在确定型图灵机上、在多项式空间内可以判定的语言类。 $PSPACE = \bigcup SPACE(n^k)$
- NPSPACE: 在非确定性图灵机上、在多项式空间内可判定的语言类。 $NPSPACE = \bigcup NSPACE(n^h)$
- \blacksquare NPSPACE = PSPACE
- $P \subseteq PSPACE$
- $\blacksquare NP \subseteq NPSPACE$
- $\blacksquare NP \subseteq PSPACE$



定义3

- PSPACE: 在确定型图灵机上、在多项式空间内可以判定的语言类。 $PSPACE = \bigcup SPACE(n^k)$
- NPSPACE: 在非确定性图灵机上、在多项式空间内可判定的语言类。 $NPSPACE = \bigcup NSPACE(n^k)$
- \blacksquare NPSPACE = PSPACE
- $P \subseteq PSPACE$
- $\blacksquare NP \subseteq NPSPACE$
- $\blacksquare NP \subseteq PSPACE$



定义3

- PSPACE: 在确定型图灵机上、在多项式空间内可以判 定的语言类。 $PSPACE = \bigcup SPACE(n^k)$
- NPSPACE: 在非确定性图灵机上、在多项式空间内可判
- $\blacksquare NPSPACE = PSPACE$
- $P \subseteq PSPACE$
- $\blacksquare NP \subseteq NPSPACE$
- $\blacksquare NP \subseteq PSPACE$

- PSPACE: 在确定型图灵机上、在多项式空间内可以判定的语言类。 $PSPACE = \bigcup_{k} SPACE(n^{k})$
- NPSPACE: 在非确定性图灵机上、在多项式空间内可判定的语言类。 $NPSPACE = \bigcup NSPACE(n^k)$
- \blacksquare NPSPACE = PSPACE
- $P \subseteq PSPACE$
- $\blacksquare NP \subseteq NPSPACE$
- $\blacksquare NP \subseteq PSPACE$



- PSPACE: 在确定型图灵机上、在多项式空间内可以判定的语言类。 $PSPACE = \bigcup_k SPACE(n^k)$
- NPSPACE: 在非确定性图灵机上、在多项式空间内可判定的语言类。 $NPSPACE = \bigcup_k NSPACE(n^k)$
- \blacksquare NPSPACE = PSPACE
- $P \subseteq PSPACE$
- \blacksquare $NP \subseteq NPSPACE$
- $\blacksquare NP \subseteq PSPACE$



- PSPACE: 在确定型图灵机上、在多项式空间内可以判定的语言类。 $PSPACE = \bigcup_k SPACE(n^k)$
- NPSPACE: 在非确定性图灵机上、在多项式空间内可判定的语言类。 $NPSPACE = \bigcup_k NSPACE(n^k)$
- \blacksquare NPSPACE = PSPACE
- $P \subseteq PSPACE$
- \blacksquare $NP \subseteq NPSPACE$
- $\blacksquare NP \subseteq PSPACE$



- PSPACE: 在确定型图灵机上、在多项式空间内可以判定的语言类。 $PSPACE = \bigcup_k SPACE(n^k)$
- NPSPACE: 在非确定性图灵机上、在多项式空间内可判定的语言类。 $NPSPACE = \bigcup_k NSPACE(n^k)$
- ightharpoonup NPSPACE = PSPACE
- $P \subseteq PSPACE$
- \blacksquare $NP \subseteq NPSPACE$
- $\blacksquare NP \subseteq PSPACE$



- PSPACE: 在确定型图灵机上、在多项式空间内可以判定的语言类。 $PSPACE = \bigcup_{k} SPACE(n^{k})$
- NPSPACE: 在非确定性图灵机上、在多项式空间内可判定的语言类。 $NPSPACE = \bigcup_k NSPACE(n^k)$
- ightharpoonup NPSPACE = PSPACE
- $P \subseteq PSPACE$
- $\blacksquare NP \subseteq NPSPACE$
- $\blacksquare NP \subseteq PSPACE$



- PSPACE: 在确定型图灵机上、在多项式空间内可以判定的语言类。 $PSPACE = \bigcup_{k} SPACE(n^{k})$
- NPSPACE: 在非确定性图灵机上、在多项式空间内可判定的语言类。 $NPSPACE = \bigcup_k NSPACE(n^k)$
- ightharpoonup NPSPACE = PSPACE
- $P \subseteq PSPACE$
- $\blacksquare NP \subseteq NPSPACE$
- $\blacksquare NP \subseteq PSPACE$



- 消耗f(n)空间的TM的格局数至多有: f(n)2O(f(n))
- 消耗空间f(n)的图灵机必定在时间 $f(n)2^{O(f(n))}$ 时间内运行
- $PSPACE \subseteq EXPTIME = \bigcup_{k} TIME(2^{n^k})$
- $P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME$
- $P \neq EXPTIME$

- 消耗f(n)空间的TM的格局数至多有: $f(n)2^{O(f(n))}$
- 消耗空间f(n)的图灵机必定在时间 $f(n)2^{O(f(n))}$ 时间内运行
- $PSPACE \subseteq EXPTIME = \bigcup_{k} TIME(2^{n^k})$
- $P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME$
- $P \neq EXPTIME$

- 消耗f(n)空间的TM的格局数至多有: $f(n)2^{O(f(n))}$
- 消耗空间f(n)的图灵机必定在时间 $f(n)2^{O(f(n))}$ 时间内运行
- $PSPACE \subseteq EXPTIME = \bigcup_{k} TIME(2^{n^k})$
- $P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIMI$
- $P \neq EXPTIME$

- 消耗f(n)空间的TM的格局数至多有: $f(n)2^{O(f(n))}$
- 消耗空间f(n)的图灵机必定在时间 $f(n)2^{O(f(n))}$ 时间内运行
- $PSPACE \subseteq EXPTIME = \bigcup_{k} TIME(2^{n^k})$
- $P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIMI$
- $P \neq EXPTIME$

- 消耗f(n)空间的TM的格局数至多有: $f(n)2^{O(f(n))}$
- 消耗空间f(n)的图灵机必定在时间 $f(n)2^{O(f(n))}$ 时间内运行
- $PSPACE \subseteq EXPTIME = \bigcup_{k} TIME(2^{n^k})$
- $\blacksquare \ P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME$
- $\blacksquare P \neq EXPTIME$

- 消耗f(n)空间的TM的格局数至多有: $f(n)2^{O(f(n))}$
- 消耗空间f(n)的图灵机必定在时间 $f(n)2^{O(f(n))}$ 时间内运行
- $PSPACE \subseteq EXPTIME = \bigcup_{k} TIME(2^{n^k})$
- $P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME$
- $P \neq EXPTIME$

若语言B满足:

- B属于PSPACE;
- PSPACE难的 PSPACE中的每个语言A都可以多项式时间归约到B

则称B是PSPACE完全的。

TQBF问题:



若语言B满足:

- B属于PSPACE;
- PSPACE难的 PSPACE中的每个语言A都可以多项式时间归约到B

则称B是PSPACE完全的。

TQBF问题:



若语言B满足:

- B属于PSPACE;
- PSPACE难的 PSPACE中的每个语言A都可以多项式时间归约到B

则称B是PSPACE完全的。

TQBF问题:



若语言B满足:

- B属于PSPACE;
- PSPACE难的 PSPACE中的每个语言A都可以多项式时 间归约到B

则称B是PSPACE完全的。



若语言B满足:

- B属于PSPACE;
- PSPACE难的 PSPACE中的每个语言A都可以多项式时间归约到B

则称B是PSPACE完全的。

TQBF问题:



若语言B满足:

- B属于PSPACE;
- PSPACE难的 PSPACE中的每个语言A都可以多项式时 间归约到B

则称B是PSPACE完全的。

TQBF问题:



■ $TQBF \in PSPACE$:

对于输入的全量词布尔公式 ϕ ,T执行

- 如果φ不含量词,即其只是常数表达式,计算φ的值
 - 如果 ϕ 的值为真,则接受;
 - 否则拒绝
- $\overline{A}\phi = \exists x\psi, 则 \overline{A}\psi$ 上递归调用T
 - 先用0替换x,再用1替换x
 - 如果至少有一个接受,则接受,省则拒绝。
- $\overline{A}\phi = \forall x\psi$,则在 ψ 上递归调用T
 - 先用0替换x,再用1替换x
 - 如果两个都为1,则接受,否则拒绝。

- $TQBF \in PSPACE$: 对于输入的全量词布尔公式 ϕ,T 执行
 - 如果ø不含量词,即其只是常数表达式,计算ø的值
 - 如果φ的值为具,则接受;
 - 否则拒绝
 - - 先用0替换x,再用1替换x
 - 如果至少有一个接受,则接受,省则拒绝。
 - $\overline{A}\phi = \forall x\psi$,则在 ψ 上递归调用T
 - 先用0替换x,再用1替换x
 - 如果两个都为1,则接受, 否则拒绝。



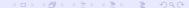
- $TQBF \in PSPACE$: 对于输入的全量词布尔公式 ϕ,T 执行
 - 如果φ不含量词,即其只是常数表达式,计算φ的值
 - 如果φ的值为真,则接受;
 - 否则拒绝
 - $\overline{A}\phi = \exists x\psi$,则在 ψ 上递归调用T
 - 先用0替换x,再用1替换x
 - ■ 如果至少有一个接受,则接受,否则拒绝.
 - $\overline{A} = \forall x \psi$,则在 ψ 上递归调用T

 - 如果两个都为1,则接受,否则拒绝。
 - 4 ロ b 4 周 b 4 国 b 4 国 b 9 Q Q

- $TQBF \in PSPACE$: 对于输入的全量词布尔公式 ϕ,T 执行
 - 如果∅不含量词,即其只是常数表达式,计算∅的值
 - 如果φ的值为真,则接受;
 - 否则拒绝



- $TQBF \in PSPACE$: 对于输入的全量词布尔公式 ϕ,T 执行
 - 如果∅不含量词,即其只是常数表达式,计算∅的值
 - 如果φ的值为真,则接受;
 - 否则拒绝
 - $\dot{\mathbf{z}}_{\phi} = \exists x \psi$,则在 ψ 上递归调用T



- $TQBF \in PSPACE$: 对于输入的全量词布尔公式 ϕ,T 执行
 - 如果φ不含量词,即其只是常数表达式,计算φ的值
 - 如果φ的值为真,则接受;
 - 否则拒绝
 - 若 $\phi = \exists x \psi$,则在 ψ 上递归调用T
 - 先用0替换x,再用1替换x
 - 如果至少有一个接受,则接受,否则拒绝。
 - 若 $\phi = \forall x \psi$,则在 ψ 上递归调用T
 - 先用0替换x,再用1替换x
 - 加里两个郏为1 则埃爲 否则指例
 - 如果两个都为1,则接受,否则拒绝。



- $TQBF \in PSPACE$: 对于输入的全量词布尔公式 ϕ,T 执行
 - 如果φ不含量词,即其只是常数表达式,计算φ的值
 - 如果φ的值为真,则接受;
 - 否则拒绝
 - 若 $\phi = \exists x \psi$,则在 ψ 上递归调用T
 - 先用0替换x,再用1替换x
 - 如果至少有一个接受,则接受,否则拒绝。
 - 若 $\phi = \forall x \psi$,则在 ψ 上递归调用T
 - 兀用U省殃x, 再
 - 如果两个都为1.则接受,否则拒绝。
- イロト イラト イミト ヨー かなべ

- $TQBF \in PSPACE$: 对于输入的全量词布尔公式 ϕ,T 执行
 - 如果φ不含量词,即其只是常数表达式,计算φ的值
 - 如果φ的值为真,则接受;
 - 否则拒绝
 - 若 $\phi = \exists x \psi$,则在 ψ 上递归调用T
 - 先用0替换x,再用1替换x
 - 如果至少有一个接受,则接受,否则拒绝。
 - 若 $\phi = \forall x \psi$,则在 ψ 上递归调用T
 - 先用0替换x,再用1替换x
 - 如果两个都为1,则接受,否则拒绝。



- $TQBF \in PSPACE$: 对于输入的全量词布尔公式 ϕ,T 执行
 - 如果φ不含量词,即其只是常数表达式,计算φ的值
 - 如果φ的值为真,则接受;
 - 否则拒绝
 - 若 $\phi = \exists x \psi$,则在 ψ 上递归调用T
 - 先用0替换x,再用1替换x
 - 如果至少有一个接受,则接受,否则拒绝。
 - 若 $\phi = \forall x \psi$,则在 ψ 上递归调用T
 - 先用0替换x,再用1替换x
 - 如果两个都为1,则接受,否则拒绝。



- $TQBF \in PSPACE$: 对于输入的全量词布尔公式 $\phi.T$ 执行
 - 如果φ不含量词,即其只是常数表达式,计算φ的值
 - 如果φ的值为真,则接受;
 - 否则拒绝
 - 若 $\phi = \exists x \psi$,则在 ψ 上递归调用T
 - 先用0替换x,再用1替换x
 - 如果至少有一个接受,则接受,否则拒绝。
 - 若 $\phi = \forall x \psi$,则在 ψ 上递归调用T
 - 先用0替换x,再用1替换x
 - 如果两个都为1,则接受, 否则拒绝。



- 对于TM M在 $O(n^k)$ 空间内判定的语言A,给 出A到TQBF的多项式时间约化
- 将输入字符串w映射为量词化布尔公式 ϕ ,使得 ϕ 为真的 充要条件是M接受w
- 利用两个代表格局的变量集合 c_1, c_2 以及t > 0,构造公式 $\phi_{c_1,c_2,t}$ 。

- 对于TM M在 $O(n^k)$ 空间内判定的语言A,给 出A到TQBF的多项式时间约化
- 将输入字符串w映射为量词化布尔公式 ϕ ,使得 ϕ 为真的 充要条件是M接受w
- 利用两个代表格局的变量集合 c_1, c_2 以及t > 0,构造公式 $\phi_{c_1,c_2,t}$ 。

- 对于TM M在 $O(n^k)$ 空间内判定的语言A,给 出A到TQBF的多项式时间约化
- 将输入字符串w映射为量词化布尔公式 ϕ ,使得 ϕ 为真的 充要条件是M接受w
- 利用两个代表格局的变量集合 c_1, c_2 以及t > 0,构造公式 ϕ_{c_1,c_2,t_3} 。

- 对于TM M在 $O(n^k)$ 空间内判定的语言A,给 出A到TQBF的多项式时间约化
- 将输入字符串w映射为量词化布尔公式 ϕ ,使得 ϕ 为真的 充要条件是M接受w
- 利用两个代表格局的变量集合 c_1, c_2 以及t > 0,构造公式 $\phi_{c_1,c_2,t}$ 。

- 每个格局有 n^k 个单元,所有有 $O(n^k)$ 个变量。
- \blacksquare 当t=1时,
 - 代表c1的每个变量与代表c2的相应变量包含相同的布尔值
 - 或利用库克-列文定理中的技巧,构造布尔表达式
- \blacksquare 当t > 1时,令

$$\phi_{c_1,c_2,t} = \exists m_1 \forall (c_3,c_4) \in \{(c_1,m_1),(m_1,c_2)\} [\phi_{c_3,c_4,t/2}]$$

- 每次递归增加长度为O(f(n))
- 递归 $\log(2^{df(n)}) = O(f(n))$ 次



- 每个格局有n^k个单元, 所有有O(n^k)个变量。
- \blacksquare 当t=1时,
 - 代表c₁的每个变量与代表c₂的相应变量包含相同的布尔值
 - 或利用库克-列文定理中的技巧,构造布尔表达式
- 当t > 1时, 令

$$\phi_{c_1,c_2,t} = \exists m_1 \forall (c_3,c_4) \in \{(c_1,m_1),(m_1,c_2)\} [\phi_{c_3,c_4,t/2}]$$

- 每次递归增加长度为O(f(n))
- 递归 $\log(2^{df(n)}) = O(f(n))$ 次



- 每个格局有*n*^k个单元, 所有有*O*(*n*^k)个变量。
- \blacksquare 当t=1时,
 - 代表c₁的每个变量与代表c₂的相应变量包含相同的布尔值
 - 或利用库克-列文定理中的技巧,构造布尔表达式
- \blacksquare 当t > 1时,令

$$\phi_{c_1,c_2,t} = \exists m_1 \forall (c_3,c_4) \in \{(c_1,m_1),(m_1,c_2)\} [\phi_{c_3,c_4,t/2}]$$

- 每次递归增加长度为O(f(n))
- 递归 $\log(2^{df(n)}) = O(f(n))$ 次

- 每个格局有*n*^k个单元, 所有有*O*(*n*^k)个变量。
- \blacksquare 当t=1时,
 - 代表c₁的每个变量与代表c₂的相应变量包含相同的布尔值
 - 或利用库克-列文定理中的技巧,构造布尔表达式
- \blacksquare 当t > 1时,令

$$\phi_{c_1,c_2,t} = \exists m_1 \forall (c_3,c_4) \in \{(c_1,m_1),(m_1,c_2)\} [\phi_{c_3,c_4,t/2}]$$

- 每次递归增加长度为O(f(n))
- **递**归 $\log(2^{df(n)}) = O(f(n))$ 次

- 每个格局有n^k个单元, 所有有O(n^k)个变量。
- \blacksquare 当t=1时,
 - 代表c₁的每个变量与代表c₂的相应变量包含相同的布尔值
 - 或利用库克-列文定理中的技巧,构造布尔表达式
- 当*t* > 1时,令

$$\phi_{c_1,c_2,t} = \exists m_1 \forall (c_3,c_4) \in \{(c_1,m_1),(m_1,c_2)\} [\phi_{c_3,c_4,t/2}]$$

- 每次递归增加长度为O(f(n))
- 递归 $\log(2^{df(n)}) = O(f(n))$ 次

- 每个格局有n^k个单元, 所有有O(n^k)个变量。
- \blacksquare 当t=1时,
 - 代表c₁的每个变量与代表c₂的相应变量包含相同的布尔值
 - 或利用库克-列文定理中的技巧,构造布尔表达式
- 当*t* > 1时,令

$$\phi_{c_1,c_2,t} = \exists m_1 \forall (c_3,c_4) \in \{(c_1,m_1),(m_1,c_2)\} [\phi_{c_3,c_4,t/2}]$$

- 每次递归增加长度为O(f(n))
- 递归 $\log(2^{df(n)}) = O(f(n))$ 次



- 每个格局有*n*^k个单元, 所有有*O*(*n*^k)个变量。
- \blacksquare 当t=1时,
 - 代表c₁的每个变量与代表c₂的相应变量包含相同的布尔值
 - 或利用库克-列文定理中的技巧,构造布尔表达式
- 当*t* > 1时,令

$$\phi_{c_1,c_2,t} = \exists m_1 \forall (c_3,c_4) \in \{(c_1,m_1),(m_1,c_2)\} [\phi_{c_3,c_4,t/2}]$$

- 每次递归增加长度为O(f(n))
- 递归 $\log(2^{df(n)}) = O(f(n))$ 次

