空间复杂度

吴　铤

2017 年 12 月 24 日

空间复杂度

空间复杂度

空间复杂度

定义1 (空间复杂度)

令M是在所有输入上都停机的确定型图灵机，M的**空间复杂度**是函数$f:\mathbb{N}\rightarrow\mathbb{N}$，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$f:\mathbb{N}\rightarrow\mathbb{R}^{+}$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$\}$
- $NSPACE(f(n))=\{L:$ L是被$O(f(n))$空间的非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$\}$
- $NSPACE(f(n))=\{L:$ L是被$O(f(n))$空间的非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$\}$
- $NSPACE(f(n))=\{L:$ L是被$O(f(n))$空间的非确定性图灵机判定的语言$\}$

例1 (SAT问题)

- 对于输入的$\langle\phi\rangle$,ϕ是布尔公式
  1. 对于ϕ中的变量$x_1,\cdots,x_m$的每个赋值
  2. 若ϕ的值为1，则接受；否则拒绝。

SAT问题的空间复杂度为$O(m)$。

例1 (SAT问题)

- 对于输入的$\langle \phi \rangle$,ϕ是布尔公式
  1. 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。

SAT问题的空间复杂度为$O(m)$。

例1 (SAT问题)

- 对于输入的$\langle\phi\rangle$,ϕ是布尔公式
  1. 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。

SAT问题的空间复杂度为$O(m)$。

例1 (SAT问题)

- 对于输入的$\langle\phi\rangle$,ϕ是布尔公式
  1. 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。

SAT问题的空间复杂度为$O(m)$。

例1 (SAT问题)

- 对于输入的$\langle\phi\rangle$,ϕ是布尔公式
  1. 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。

SAT问题的空间复杂度为$O(m)$。

定理1 (萨维奇定理)

对于任意满足$f(n) \geq n$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

定理1 (萨维奇定理)

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

定理1 (萨维奇定理)

对于任意满足$f(n) \geq n$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

定理1 (萨维奇定理)

对于任意满足$f(n) \geq n$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

CANYIELD(c_1, c_2, t):

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
3. 若还没有接受，则拒绝。

CANYIELD(c_1, c_2, t):

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
3. 若还没有接受，则拒绝。

CANYIELD(c_1, c_2, t):

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
3. 若还没有接受，则拒绝。

CANYIELD(c_1, c_2, t):

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
3. 若还没有接受，则拒绝。

CANYIELD(c_1, c_2, t):

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
3. 若还没有接受，则拒绝。

CANYIELD(c_1, c_2, t):

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
3. 若还没有接受，则拒绝。

CANYIELD(c_1, c_2, t):

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
3. 若还没有接受，则拒绝。

CANYIELD(c_1, c_2, t):

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
3. 若还没有接受，则拒绝。

CANYIELD(c_1, c_2, t):

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
3. 若还没有接受，则拒绝。

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$(c_{start}, c_{accept}, 2^{df(n)})$
- 递归的每一层增加的空间：$O(f(n))$
- 递归次数：$O(\log 2^{df(n)}) = O(f(n))$

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$(c_{start}, c_{accept}, 2^{df(n)})$
- 递归的每一层增加的空间：$O(f(n))$
- 递归次数：$O(\log 2^{df(n)}) = O(f(n))$

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$(c_{start}, c_{accept}, 2^{df(n)})$
- 递归的每一层增加的空间：$O(f(n))$
- 递归次数：$O(\log 2^{df(n)}) = O(f(n))$

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$(c_{start}, c_{accept}, 2^{df(n)})$
- 递归的每一层增加的空间：$O(f(n))$
- 递归次数：$O(\log 2^{df(n)}) = O(f(n))$

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$(c_{start}, c_{accept}, 2^{df(n)})$
- 递归的每一层增加的空间：$O(f(n))$
- 递归次数：$O(\log 2^{df(n)}) = O(f(n))$

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$(c_{start}, c_{accept}, 2^{df(n)})$
- 递归的每一层增加的空间：$O(f(n))$
- 递归次数：$O(\log 2^{df(n)}) = O(f(n))$

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$(c_{start}, c_{accept}, 2^{df(n)})$
- 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。$PSPACE = \bigcup_k SPACE(n^k)$
- *NPSPACE*: 在非确定性图灵机上、在多项式空间内可判定的语言类。$NPSPACE = \bigcup_k NSPACE(n^k)$
- $NPSPACE = PSPACE$
- $P \subseteq PSPACE$
- $NP \subseteq NPSPACE$
- $NP \subseteq PSPACE$

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

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

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

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

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

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

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

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

- 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
- $PSPACE \subseteq EXPTIME = \bigcup_k TIME(2^{n^k})$
- $P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME$
- $P \neq EXPTIME$

- 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
- $PSPACE \subseteq EXPTIME = \bigcup_k TIME(2^{n^k})$
- $P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME$
- $P \neq EXPTIME$

- 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
- $PSPACE \subseteq EXPTIME = \bigcup_k TIME(2^{n^k})$
- $P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME$
- $P \neq EXPTIME$

定义4 (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:

$$TQBF = \{\langle\phi\rangle : \phi\text{是真的全量词化的布尔公式}\}$$

定义4 (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:

$$TQBF = \{\langle\phi\rangle : \phi\text{是真的全量词化的布尔公式}\}$$

定义4 (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:

$$TQBF = \{\langle\phi\rangle : \phi\text{是真的全量词化的布尔公式}\}$$

定义4 (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:

$$TQBF = \{\langle\phi\rangle : \phi\text{是真的全量词化的布尔公式}\}$$

定义4 (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:

$$TQBF = \{\langle\phi\rangle : \phi\text{是真的全量词化的布尔公式}\}$$

定义4 (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:

$$TQBF = \{\langle \phi \rangle : \phi\text{是真的全量词化的布尔公式}\}$$

- $TQBF \in PSPACE$:
  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是PSPACE难的：

- 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$\phi_{c_1,c_2,t}$。

TQBF是PSPACE难的：

- 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$\phi_{c_1,c_2,t}$。

TQBF是PSPACE难的：

- 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$\phi_{c_1,c_2,t}$。

TQBF是PSPACE难的：

- 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。
- 当$t=1$时，
  - 代表c_1的每个变量与代表c_2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。
- 当$t=1$时，
  - 代表c_1的每个变量与代表c_2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。
- 当$t = 1$时，
  - 代表c_1的每个变量与代表c_2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。
- 当$t=1$时，
  - 代表c_1的每个变量与代表c_2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。
- 当$t=1$时，
  - 代表c_1的每个变量与代表c_2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。
- 当$t = 1$时，
  - 代表c_1的每个变量与代表c_2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。
- 当$t=1$时，
  - 代表c_1的每个变量与代表c_2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