Building Algorithmic Polytopes

David Bremner

with D. Avis, H.R. Tiwary, and O. Watanabe

December 16, 2014

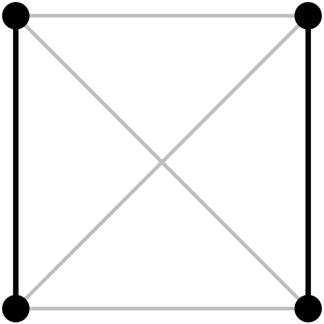
Outline

Matchings in graphs

Given $G = (V, E)$, $M \subset E$ is called a *matching*

$$|\{e \in M \mid v \in e\}| \leq 1 \quad \forall v \in V$$

A matching M is a *perfect matching* if $|M| = |V|/2$



Edmonds' Matching Polytope

Convex Hull Description

$$\mathrm{EM}_n = \mathrm{conv}\{\, \chi(M) \in \{0,1\}^{\binom{n}{2}} \mid M \text{ matching in } K_n\}$$

Inequality Description

$$\begin{aligned} x_e &\geq 0 && e \in E \\ \sum_{e \ni v} x_e &\leq 1 && v \in V \\ \sum_{e \subset W} x_e &\leq (|W|-1)/2 && W \subset V,\ |W| \text{ odd} \end{aligned}$$

Extended Formulation

Definition

An *extended formulation* (EF) of a polytope $P \subseteq \mathbb{R}^d$ is a linear system

$$Ex + Fy = g, \;\; y \geqslant \mathbf{0}$$

such that $P = \{x \mid \exists y \; Ex + Fy = g\}$

Extended Formulation

Definition

An *extended formulation* (EF) of a polytope $P \subseteq \mathbb{R}^d$ is a linear system

$$Ex + Fy = g, \ \ y \geqslant \mathbf{0}$$

such that $P = \{x \mid \exists y \; Ex + Fy = g\}$

Theorem (Rothvoß2013)

Any extended formulation of Edmonds' matching polytope EM_n *has* $2^{\Omega(n)}$ *inequalities.*

Outline

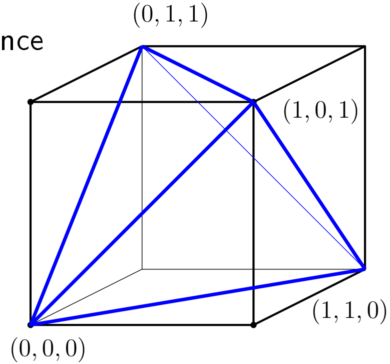
Polytopes for decision problems

Consider a decision problem defined by its *characteristic function*

$$\psi(x) = \begin{cases} 1 & x \text{ char. vec. of } \textbf{YES} \text{ instance} \\ 0 & \text{otherwise} \end{cases}$$

For each input size q we can define a polytope

$$P(\psi, q) = \operatorname{conv}\{(x, \psi(x)) : x \in \{0,1\}^q\}$$

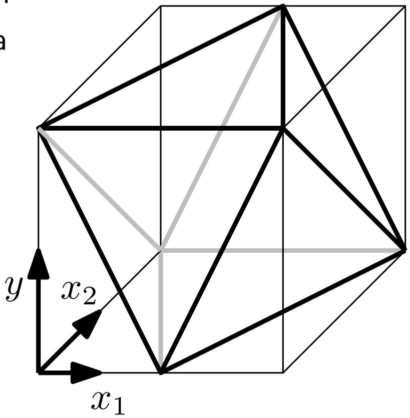


0/1-property

Definition

Let $Q \subseteq [0,1]^{q+t}$ be a polytope. We say that Q has the *x-0/1 property* if

- For each x in $\{0,1\}^q$ there is a unique vertex (x,y) of Q, and
- $(x,y) \in \{0,1\}^{q+t}$.

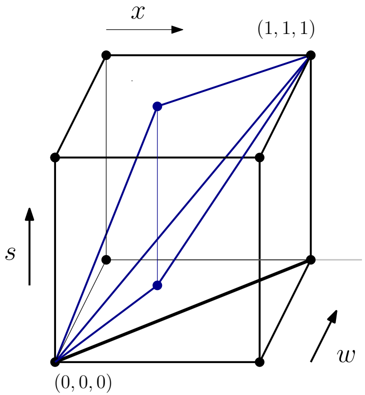


Weak Extended Formulation

Let $Q \subseteq [0,1]^{q+1+r}$. $\forall \bar{x} \in \{0,1\}^q$, $0 < \delta \leq 1/2$, define $c_i = (2\bar{x}_i - 1)$,

$$z^* = \max \sum_i c_i x_i + \delta w - \mathbb{1}^T \bar{x} \qquad \text{(LP)}$$
$$(x, w, s) \in Q$$

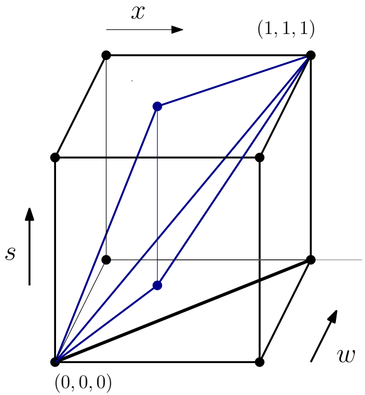
Q is a *weak extended formulation (WEF)* of $P(\psi, q)$ if Q has the x-0/1 property, and



Weak Extended Formulation

Let $Q \subseteq [0,1]^{q+1+r}$. $\forall \bar{x} \in \{0,1\}^q$, $0 < \delta \leq 1/2$, define $c_i = (2\bar{x}_i - 1)$,

$$z^* = \max \sum_i c_i x_i + \delta w - \mathbb{1}^T \bar{x} \qquad \text{(LP)}$$
$$(x, w, s) \in Q$$



Q is a *weak extended formulation (WEF)* of $P(\psi, q)$ if Q has the x-0/1 property, and

- If $\psi(\bar{x}) = 1$ the solution to (??) is unique and $z^* = \delta$.

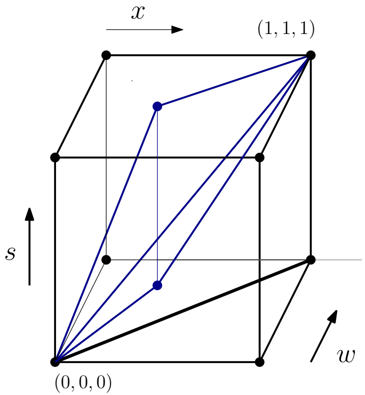
Weak Extended Formulation

Let $Q \subseteq [0,1]^{q+1+r}$. $\forall \bar{x} \in \{0,1\}^q$, $0 < \delta \leq 1/2$, define $c_i = (2\bar{x}_i - 1)$,

$$z^* = \max \sum_i c_i x_i + \delta w - \mathbb{1}^T \bar{x} \qquad \text{(LP)}$$
$$(x, w, s) \in Q$$

Q is a *weak extended formulation (WEF)* of $P(\psi, q)$ if Q has the x-0/1 property, and

- If $\psi(\bar{x}) = 1$ the solution to (??) is unique and $z^* = \delta$.
- Otherwise $z^* < \delta$ and $\forall \delta \leq \epsilon(q+r)$, $z^* = 0$ and the solution to (??) is unique.



Outline

Integer Register machines (Cook and Reckhow)

Operations

- $x \leftarrow y \pm z$
- $x \leftarrow y[z]$
- $x[y] \leftarrow z$
- if $x > 0$ goto L (L is a constant line number)

Integer Register machines (Cook and Reckhow)

Operations

- $x \leftarrow y \pm z$
- $x \leftarrow y[z]$
- $x[y] \leftarrow z$
- if $x > 0$ goto L (L is a constant line number)

Register size and costs

- registers can hold arbitrarily large/small integers
- cost of operations is proportional to $\log_2$ of operand size

Binary register machines

Bounding operand sizes

- assume running time is bounded by $p(n)$
- from cost model $|x| \leq 2^{p(n)}$
- often we know $|x| \leq M \ll 2^{p(n)}$
- define a *parameter* $\beta = \log_2 M$

Binary register machines

Bounding operand sizes

- assume running time is bounded by $p(n)$
- from cost model $|x| \leq 2^{p(n)}$
- often we know $|x| \leq M \ll 2^{p(n)}$
- define a *parameter* $\beta = \log_2 M$

Binary registers

- Arbitrary number of named β-bit *integer* registers
- Arbitrary number of named *arrays* of integer registers, each containing at most 2^β elements.

Boolean registers and 2D arrays

Boolean registers

- operations on 1-bit registers turn out to be much easier
- E.g. *sets* can be represented arrays of booleans.

Boolean registers and 2D arrays

Boolean registers

- operations on 1-bit registers turn out to be much easier
- E.g. *sets* can be represented arrays of booleans.

2D arrays

- Arbitrary number of named *2D arrays* of boolean registers, containing at most $2^\beta \times 2^\beta$ elements
- handy for representing graphs

ASM code

Boolean operations

- $x \leftarrow y \circ z$
  $\circ \in \{\vee, \wedge, \oplus, =\}$
- $x \leftarrow y[j]$, $x \leftarrow y[j, k]$
- $x[i] \leftarrow z$, $x[i, j] \leftarrow y$

ASM code

Boolean operations

- $x \leftarrow y \circ z$
  $\circ \in \{\vee, \wedge, \oplus, =\}$
- $x \leftarrow y[j]$, $x \leftarrow y[j, k]$
- $x[i] \leftarrow z$, $x[i, j] \leftarrow y$

Integer operations

- $i \leftarrow j + 1$
- $x \leftarrow i = j$
- $i \leftarrow j[k]$
- $i[j] \leftarrow [k]$

ASM code

Boolean operations

- $x \leftarrow y \circ z$
  $\circ \in \{\vee, \wedge, \oplus, =\}$
- $x \leftarrow y[j]$, $x \leftarrow y[j,k]$
- $x[i] \leftarrow z$, $x[i,j] \leftarrow y$

Integer operations

- $i \leftarrow j + 1$
- $x \leftarrow i = j$
- $i \leftarrow j[k]$
- $i[j] \leftarrow [k]$

Control Flow

- if x goto L (L is a constant line number)
- goto L
- return $w@v$

Outline

Block structured imperative language

SPARKS

- named after Horowitz-Sahni FORTRAN preprocessor
- close to traditional pseudocode
- generates easy to parse ASM code

Syntax

- control flow: if-then-else/while/for
- compound expressions
- type/input/output declarations

if

```
input bool x
input bool y
output bool w
if x then
   if y then
      return w @ 1
   else
      return w @ 0
   endif
else
   return w @ 0
endif
```

```
. input bool x
. input bool y
. output bool w
. set guard0 copy x
. set guard0 not guard0
. if guard0 else0
. set guard1 copy y
. set guard1 not guard1
. if guard1 else1
2 return w copy 1
else1 nop
3 return w copy 0
else0 nop
4 return w copy 0
```

for

```
for i <- 1,3 do
    nop
done
```

```
. set i copyw 1
. set sentinel0 copyw 3
. set sentinel0 incw sentinel0
for0 set test0 eqw i sentinel0
. if test0 done0
. nop
. set i incw i
. goto for0
done0 nop
```

Outline

GMPL as target

- inequalities are output as *Gnu Math Programming Language*
- preservation of names, array structure, helps debugging
- Can be solved directly by *glpsol*, or transformed to MPS / matrix form.

Polytopes from ASM

Inspiration

- Modelled on proof of Cook's theorem from [HS-1978]
- reduction of simplified SPARKS code to Boolean SAT

Inequality groups

- C initialization
- D begin at the beginning
- E one line at a time
- F control flow
- G memory (non)-updates

Polytopes from ASM

Inspiration

- Modelled on proof of Cook's theorem from [HS-1978]
- reduction of simplified SPARKS code to Boolean SAT

Inequality groups

- C initialization
- D begin at the beginning
- E one line at a time
- F control flow
- G memory (non)-updates

Parameters

- From ASM code $\mathtt{A}(n, \beta)$, polytopes $Q(\mathtt{A}(n, \beta))$.

Adding time dimension

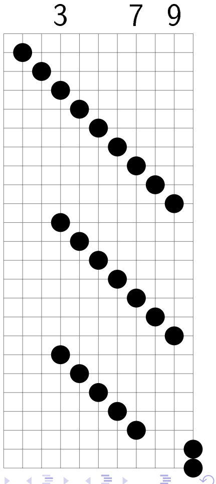
- each variable is given an extra time dimension

```
bool x             var x{0..tmax-1},>=0,<=1;
int y              var y{0..bits-1,0..tmax-1},>=0,<=1;
array A[10]        var A{0..10,0..tmax-1},>=0,<=1;
matrix M[7,7]      var M{0..7,0..7,0..tmax-1},>=0,<=1;
```

The step counter

$$S[i,t] = \begin{cases} 1 & \text{line } i \text{ of } \mathtt{A} \text{ is being executed at time } t \\ 0 & \text{otherwise} \end{cases}$$

```
1  var int i
2  set i copyw 1
3  nop
4  set temp2 eqw i 3
5  set test1 not temp2
6  set test1 not test1
7  if test1 10
8  set i incw i
9  goto 3
10 nop
```



Controlled 0/1 property

$$
\begin{aligned}
-2x + y &\leq 0 \\
2x - y &\leq 1 \\
y &\leq 1 \\
-y &\leq 0
\end{aligned}
$$

Definition

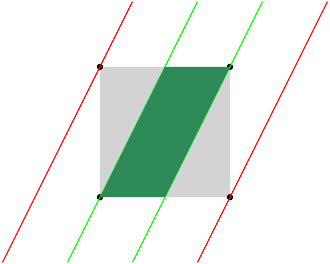
Suppose

1. $Cx + Dy \leqslant e$ has the x-0/1 property.
2. $Cx + Dy \leqslant e + \mathbb{1}$ is feasible for all $(x, y) \in \{0, 1\}^q$.

The system

$$\mathbb{1}z + Cx + Dy \leqslant e + \mathbb{1}$$

has the (z) *controlled x-0/1 property*.



basic inequalities for the step counter

(D) Step counter initialization

Instruction 1 is executed at time $t = 1$.

$$S[1,1] = 1$$

(E) Unique step execution

A unique instruction is executed at each time t.

$$\sum_{j=1}^{l} S[i,t] = 1, \qquad 1 \leqslant t \leqslant p(n)$$

(F) Inequalities for flow control

Inequalities are generated for each t, $1 \leqslant t \leqslant p(n)$, depending on the instruction at line i

(i) **(assignment statement)** Go to the next instruction.

$$S[i, t] - S[i+1, t+1] \leqslant 0$$

(F) Inequalities for flow control

Inequalities are generated for each t, $1 \leqslant t \leqslant p(n)$, depending on the instruction at line i

(i) **(assignment statement)** Go to the next instruction.

$$S[i,t] - S[i+1, t+1] \leqslant 0$$

(ii) **(go to k)**

$$S[i,t] - S[k, t+1] \leqslant 0$$

(F) Inequalities for flow control

Inequalities are generated for each t, $1 \leqslant t \leqslant p(n)$, depending on the instruction at line i

(i) **(assignment statement)** Go to the next instruction.

$$S[i,t] - S[i+1,t+1] \leqslant 0$$

(ii) **(go to k)**

$$S[i,t] - S[k,t+1] \leqslant 0$$

(iii) **(return)** Loop on this line until time runs out.

$$S[i,t] - S[i,t+1] \leqslant 0$$

(F) Inequalities for flow control

Inequalities are generated for each t, $1 \leqslant t \leqslant p(n)$, depending on the instruction at line i

(i) **(assignment statement)** Go to the next instruction.

$$S[i,t] - S[i+1,t+1] \leqslant 0$$

(ii) **(go to k)**

$$S[i,t] - S[k,t+1] \leqslant 0$$

(iii) **(return)** Loop on this line until time runs out.

$$S[i,t] - S[i,t+1] \leqslant 0$$

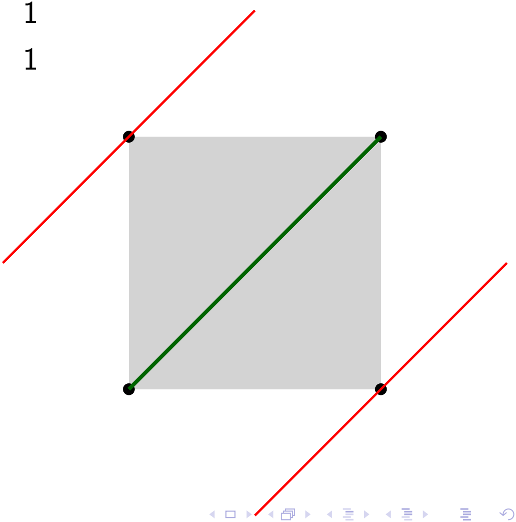
(iv) **(if c goto k)**

$$\begin{aligned} S[i,t] + c[t-1] - S[k,t+1] &\leqslant 1 \\ S[i,t] - c[t-1] - S[i+1,t+1] &\leqslant 0 \end{aligned}$$

(G) assignment: $s = x$

For $s = x$ we generate the two inequalities:

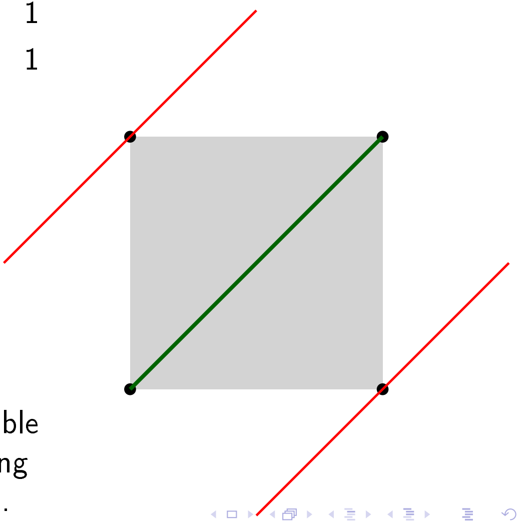
$$
\begin{aligned}
S[i,t] + x[t-1] - s[t] &\leqslant 1 \\
S[i,t] - x[t-1] + s[t] &\leqslant 1
\end{aligned}
$$



(G) assignment: $s = x$

For $s = x$ we generate the two inequalities:

$$
\begin{aligned}
S[i,t] + x[t-1] - s[t] &\leqslant 1 \\
S[i,t] - x[t-1] + s[t] &\leqslant 1
\end{aligned}
$$



- Note that 'x[t]' makes sense in place of 's[t]'
- Every unmodified variable is "carried forward" using these same inequalities.

(G) Boolean exclusive or

$s = x \oplus y$

$$\begin{aligned}
S[i,t] + x[t-1] - y[t-1] - s[t] &\leqslant 1 \\
S[i,t] - x[t-1] - y[t-1] + s[t] &\leqslant 1 \\
S[i,t] - x[t-1] + y[t-1] - s[t] &\leqslant 1 \\
S[i,t] + x[t-1] + y[t-1] + s[t] &\leqslant 3
\end{aligned}$$

(G) Boolean exclusive or

$s = x \oplus y$

$$\begin{aligned}
S[i,t] + x[t-1] - y[t-1] - s[t] &\leqslant 1 \\
S[i,t] - x[t-1] - y[t-1] + s[t] &\leqslant 1 \\
S[i,t] - x[t-1] + y[t-1] - s[t] &\leqslant 1 \\
S[i,t] + x[t-1] + y[t-1] + s[t] &\leqslant 3
\end{aligned}$$

$S[i,t] = 1$

$$+x[t-1] - y[t-1] \leqslant s[t] \tag{10}$$

$$s[t] \leqslant x[t-1] + y[t-1] \tag{00}$$

$$-x[t-1] + y[t-1] \leqslant s[t] \tag{01}$$

$$s[t] \leqslant 2 - x[t-1] - y[t-1] \tag{11}$$

Integer increment $q = q + 1$

- a second integer variable r holds the carries

$$
\begin{aligned}
q[1,t] &= q[1,t-1] \oplus 1 \\
r[1,t] &= q[1,t-1] \wedge 1 \\
r[j,t] &= q[j,t-1] \wedge r[j-1,t] \qquad 2 \leqslant j \leqslant \beta \\
q[j,t] &= q[j,t-1] \oplus r[j-1,t] \qquad 2 \leqslant j \leqslant \beta
\end{aligned}
$$

- Each of these equations is enforced with sets of inequalities

array assignment 1/2

- $x \leftarrow R[m]$, R has indicies $0..u$

Comparison representation of index m

$$\mu(j,t) = \begin{cases} 0 & m[t-1] = j \\ 1 & \text{otherwise} \end{cases}$$

$$= \bigvee_{k=1}^{\beta} m[k, t-1] \oplus \mathrm{bit}(j,k)$$

For $0 \leq j \leq u$

$$S[i,t] + \mu(j,t) - M_i[j,t] \leqslant 1$$
$$S[i,t] - \mu(j,t) + M_i[j,t] \leqslant 1$$

array assignment 2/2

inequalities

$$
\begin{aligned}
S[i,t] + x[t-1] - R[j,t] - M_i[j,t] &\leqslant 1 \\
S[i,t] - x[t-1] + R[j,t] - M_i[j,t] &\leqslant 1 \\
S[i,t] + R[j,t-1] - R[j,t] + M_i[j,t] &\leqslant 2 \\
S[i,t] - R[j,t-1] + R[j,t] + M_i[j,t] &\leqslant 2
\end{aligned}
$$

array assignment 2/2

inequalities

$$
\begin{aligned}
S[i,t] + x[t-1] - R[j,t] - M_i[j,t] &\leqslant 1 \\
S[i,t] - x[t-1] + R[j,t] - M_i[j,t] &\leqslant 1 \\
S[i,t] + R[j,t-1] - R[j,t] + M_i[j,t] &\leqslant 2 \\
S[i,t] - R[j,t-1] + R[j,t] + M_i[j,t] &\leqslant 2
\end{aligned}
$$

$S[i,t] = 1$

$$
\begin{aligned}
+x[t-1] - R[j,t] &\leqslant M_i[j,t] \\
-x[t-1] + R[j,t] &\leqslant M_i[j,t] \\
+R[j,t-1] - R[j,t] &\leqslant 1 - M_i[j,t] \\
-R[j,t-1] + R[j,t] &\leqslant 1 - M_i[j,t]
\end{aligned}
$$

array assignment 2/2

inequalities

$$
\begin{aligned}
S[i,t] + x[t-1] - R[j,t] - M_i[j,t] &\leqslant 1\\
S[i,t] - x[t-1] + R[j,t] - M_i[j,t] &\leqslant 1\\
S[i,t] + R[j,t-1] - R[j,t] + M_i[j,t] &\leqslant 2\\
S[i,t] - R[j,t-1] + R[j,t] + M_i[j,t] &\leqslant 2
\end{aligned}
$$

$S[i,t] = 1$

$$
\begin{aligned}
+x[t-1] - R[j,t] &\leqslant M_i[j,t]\\
-x[t-1] + R[j,t] &\leqslant M_i[j,t]\\
+R[j,t-1] - R[j,t] &\leqslant 1 - M_i[j,t]\\
-R[j,t-1] + R[j,t] &\leqslant 1 - M_i[j,t]
\end{aligned}
$$

Main idea

- $M_i[j,t]$ acts as a switch between two assignments

Polytopes that compute

Proposition

- *Let* $\mathtt{A}(n,\beta)$ *be an ASM code with input* $x \in [0,1]^n$ *that terminates by setting* $w = \psi(x)$.
- *Let* $Q(n,\beta)$ *be the constructed polytope with extra variables* s_i.

Then we have

Polytopes that compute

Proposition

- *Let* $\mathtt{A}(n, \beta)$ *be an ASM code with input* $x \in [0, 1]^n$ *that terminates by setting* $w = \psi(x)$.
- *Let* $Q(n, \beta)$ *be the constructed polytope with extra variables* s_i.

Then we have

1. $Q(n, \beta)$ *has size polynomial in the running time of* $\mathtt{A}$.

Polytopes that compute

Proposition

- *Let* $\mathtt{A}(n, \beta)$ *be an ASM code with input* $x \in [0, 1]^n$ *that terminates by setting* $w = \psi(x)$.
- *Let* $Q(n, \beta)$ *be the constructed polytope with extra variables* s_i.

Then we have

1. $Q(n, \beta)$ *has size polynomial in the running time of* $\mathtt{A}$.
2. *For any* $x^* \in \{0, 1\}^n$, $Q(n, \beta)$ *has a unique vertex* (x^*, w^*, s^*) *with* $w^* = \psi(x^*)$.

Polytopes that compute

Proposition

- *Let* $\mathtt{A}(n, \beta)$ *be an ASM code with input* $x \in [0, 1]^n$ *that terminates by setting* $w = \psi(x)$.
- *Let* $Q(n, \beta)$ *be the constructed polytope with extra variables* s_i.

Then we have

1. $Q(n, \beta)$ *has size polynomial in the running time of* $\mathtt{A}$.
2. *For any* $x^* \in \{0, 1\}^n$, $Q(n, \beta)$ *has a unique vertex* (x^*, w^*, s^*) *with* $w^* = \psi(x^*)$.

Proof.

By induction on timestep t. $\square$

Weak extended formulations

Proposition

Let $\mathtt{A}(n, \beta)$ *be an ASM code which solves a decision problem with characteristic function* $\psi : \{0,1\}^n \to \{0,1\}$*. The corresponding polytope* $Q(n, \beta)$ *is a weak extended formulation for* $P(\psi, n)$*.*

Weak extended formulations

Proposition

Let $\mathtt{A}(n, \beta)$ be an ASM code which solves a decision problem with characteristic function $\psi : \{0,1\}^n \to \{0,1\}$. The corresponding polytope $Q(n, \beta)$ is a weak extended formulation for $P(\psi, n)$.

- $Q(n, \beta)$ has the x-0/1 property (previous proposition)

Weak extended formulations

Proposition

Let $\mathtt{A}(n, \beta)$ be an ASM code which solves a decision problem with characteristic function $\psi : \{0,1\}^n \to \{0,1\}$. The corresponding polytope $Q(n, \beta)$ is a weak extended formulation for $P(\psi, n)$.

- $Q(n, \beta)$ has the x-0/1 property (previous proposition)
- The objective function $z(x) = \sum_i (2\bar{x}_i - 1)x_i + \delta$ forces the optimal solution $(\bar{x}, \psi(\bar{x}), s)$ for sufficiently small δ.

Weak extended formulations

Proposition

Let $\mathtt{A}(n, \beta)$ *be an ASM code which solves a decision problem with characteristic function* $\psi : \{0,1\}^n \to \{0,1\}$*. The corresponding polytope* $Q(n, \beta)$ *is a weak extended formulation for* $P(\psi, n)$*.*

- $Q(n, \beta)$ has the x-0/1 property (previous proposition)
- The objective function $z(x) = \sum_i (2\bar{x}_i - 1)x_i + \delta$ forces the optimal solution $(\bar{x}, \psi(\bar{x}), s)$ for sufficiently small δ.
- Such a δ can be computed quickly.