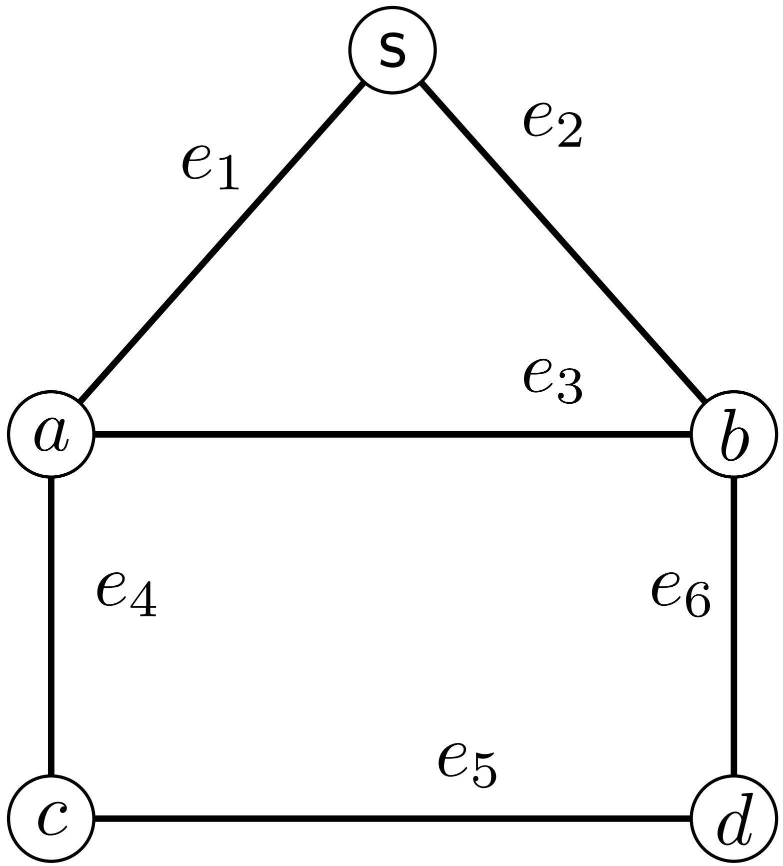
One-parameter Mechanisms

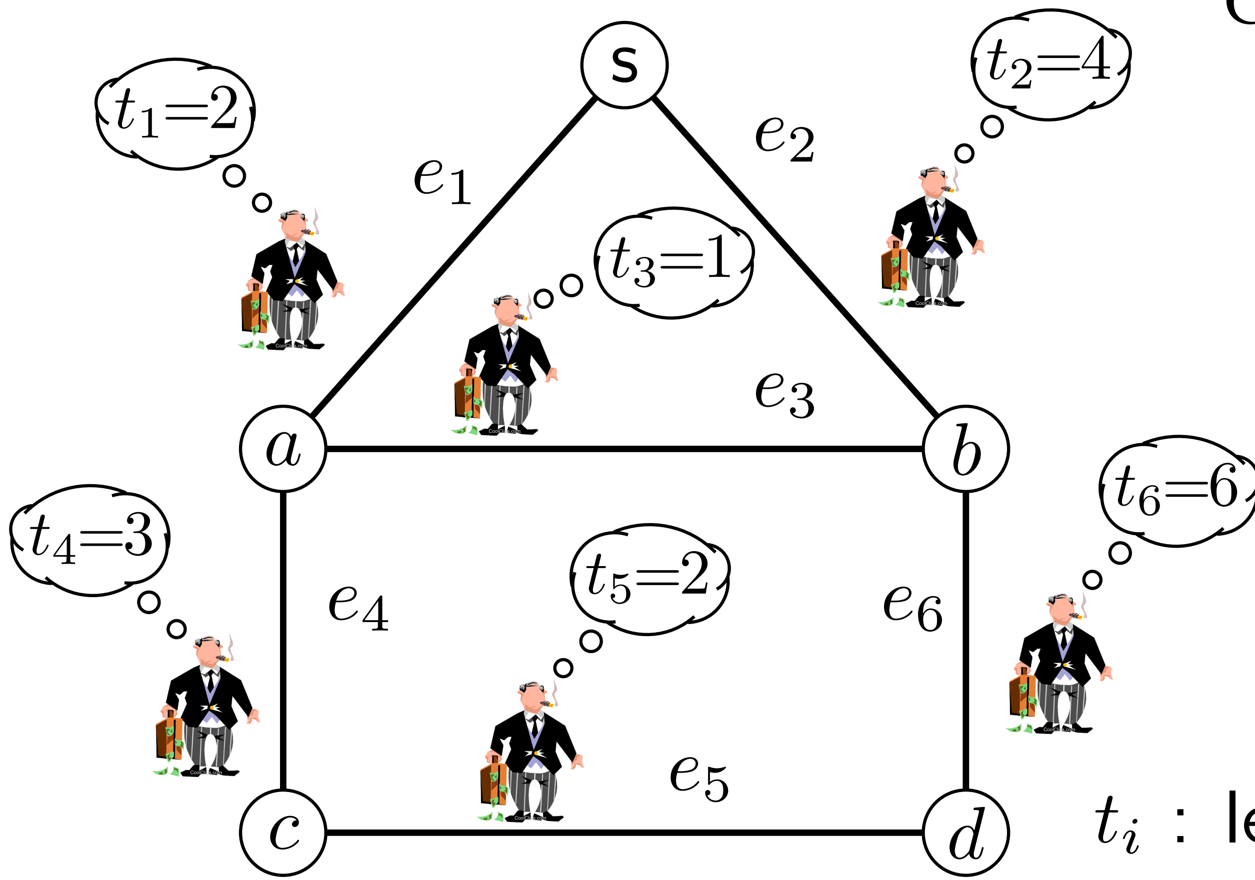
The private-edge Shortest-Paths Tree (SPT) problem



$$G = (V, E)$$

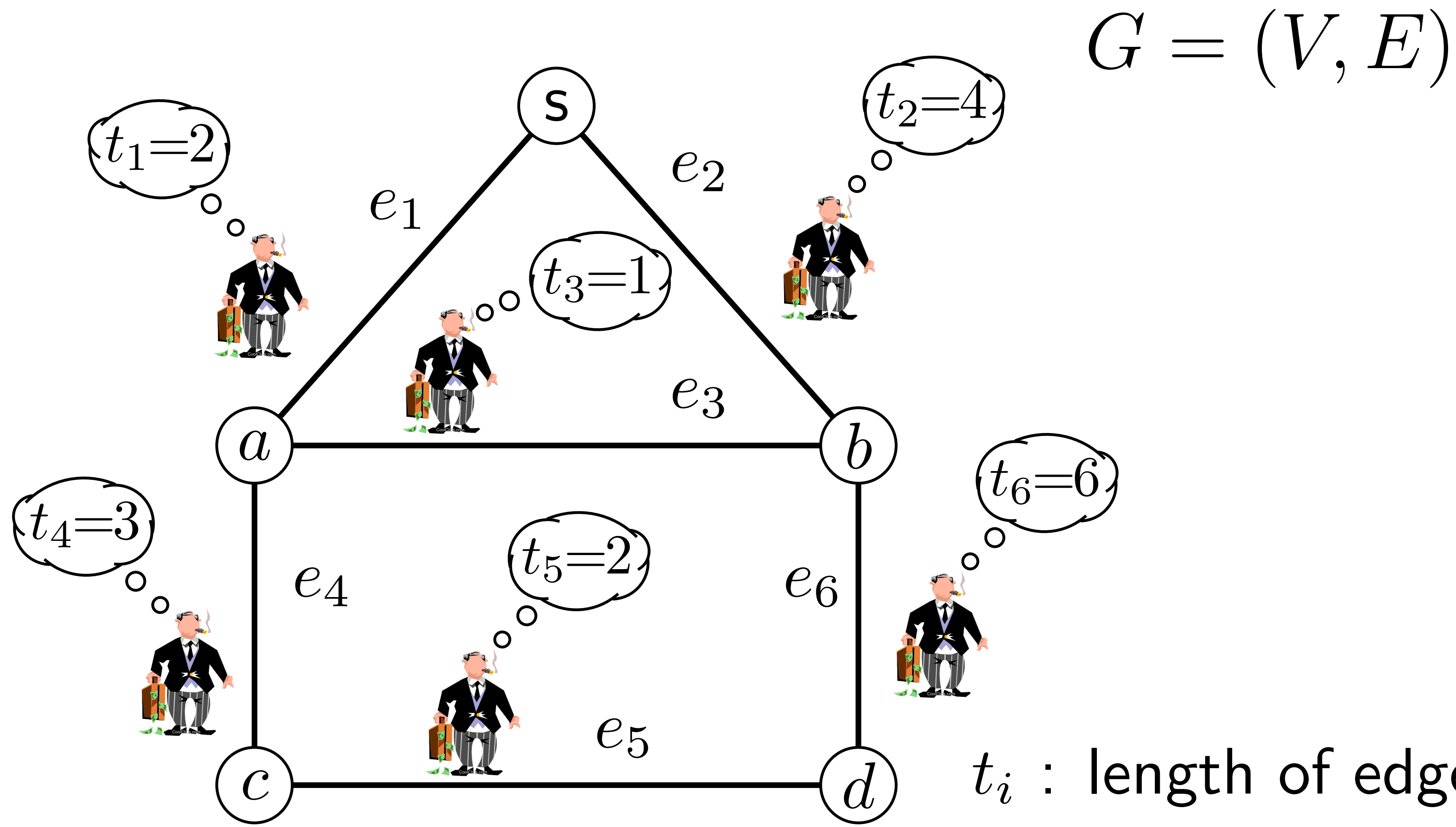
The private-edge Shortest-Paths Tree (SPT) problem

$G = (V, E)$



t_i : length of edge e_i

The private-edge Shortest-Paths Tree (SPT) problem

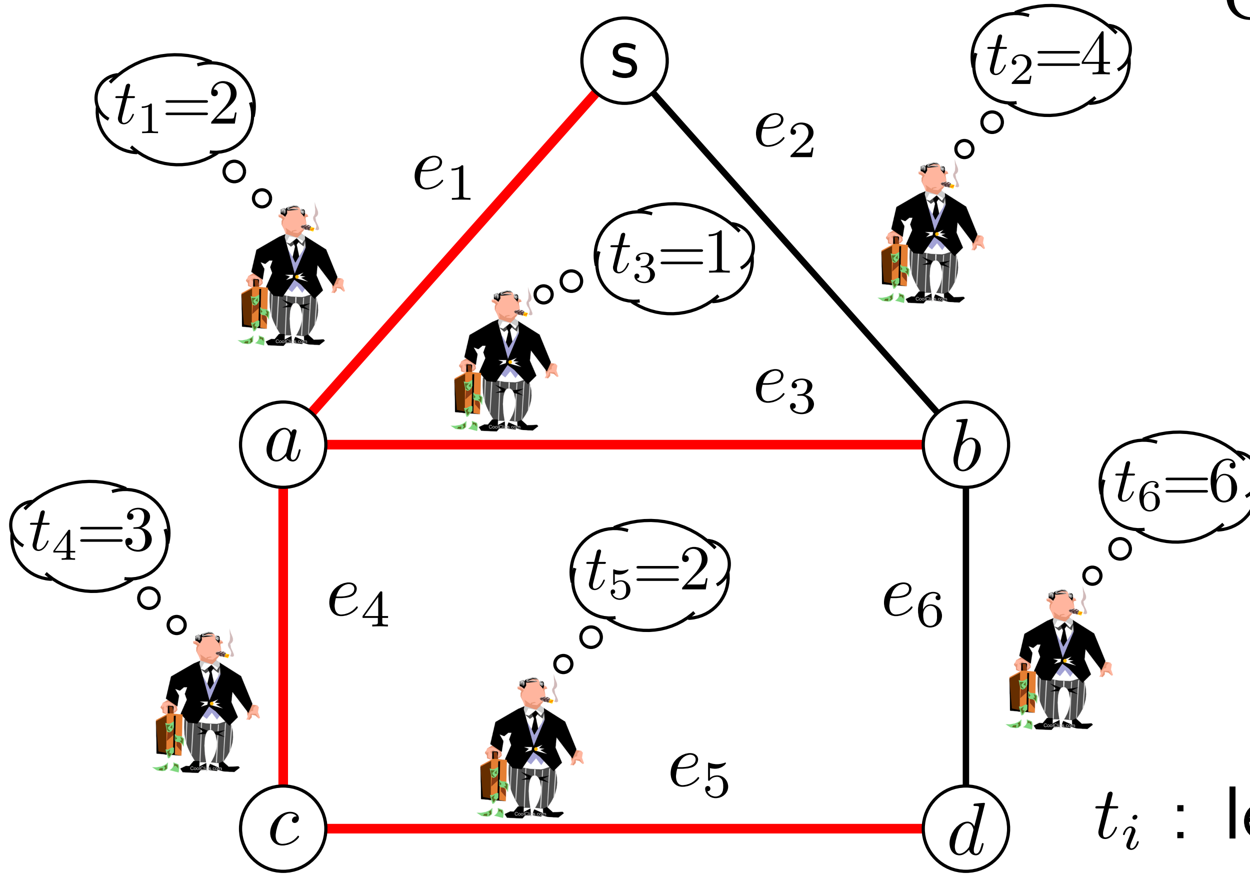


t_i : length of edge e_i

Goal: Design an efficient *truthful mechanism* to find a shortest-path tree (SPT) of G rooted at s.

The private-edge Shortest-Paths Tree (SPT) problem

$G = (V, E)$



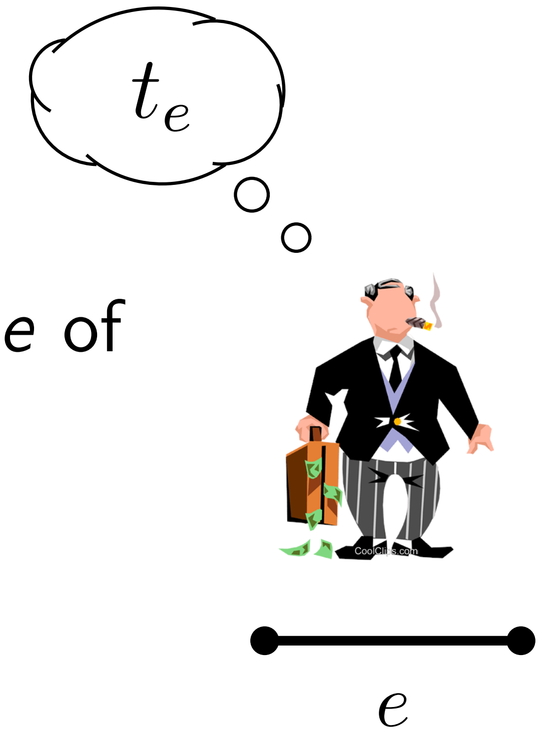
t_i : length of edge e_i

Goal: Design an efficient *truthful mechanism* to find a shortest-path tree (SPT) of G rooted at s.

The private-edge Shortest-Paths Tree (SPT) problem

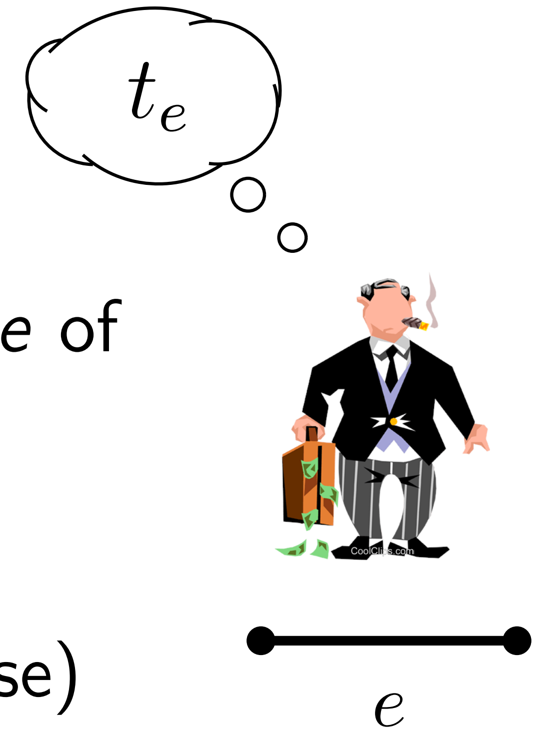
- Each edge e is owned by a selfish agent

- The *length* t_e of edge e is the *private type* of agent e



The private-edge Shortest-Paths Tree (SPT) problem

- Each edge e is owned by a selfish agent
- The *length* t_e of edge e is the *private type* of agent e
- Agent e incurs a cost of t_e if edge e is selected in the SPT (and no cost otherwise)



The private-edge Shortest-Paths Tree (SPT) problem

- Each edge e is owned by a selfish agent

- The *length* t_e of edge e is the *private type* of agent e

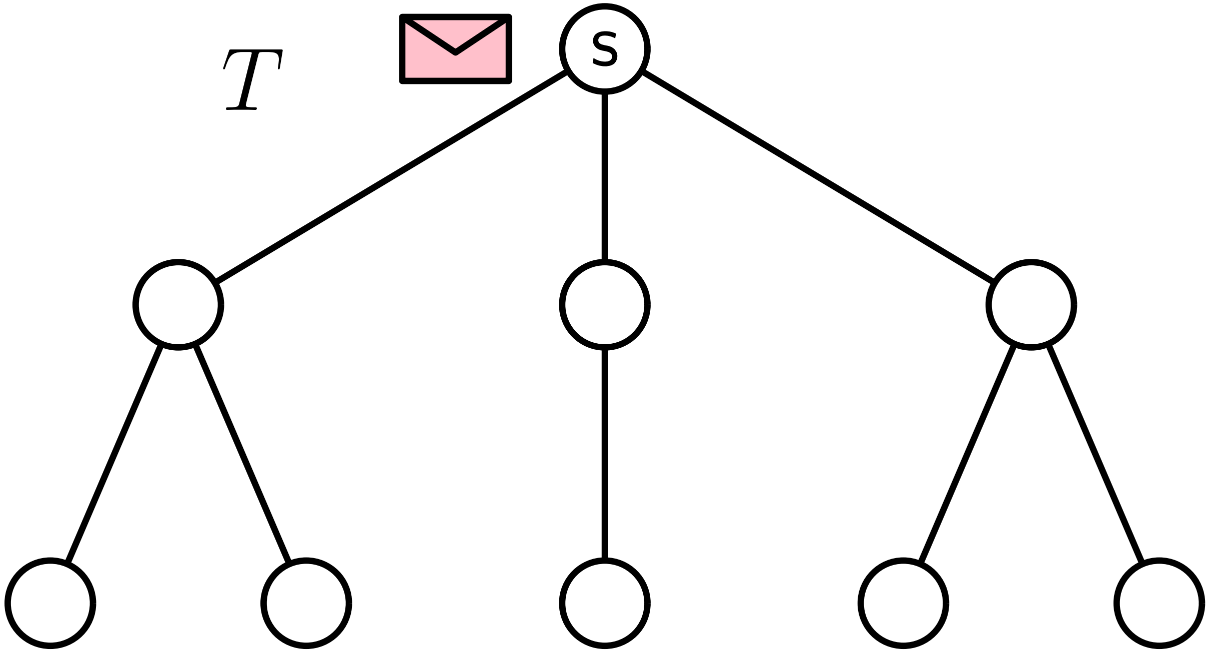
- Agent e incurs a cost of t_e if edge e is selected in the SPT (and no cost otherwise)

- The valuation of agent e w.r.t. to a tree T is:

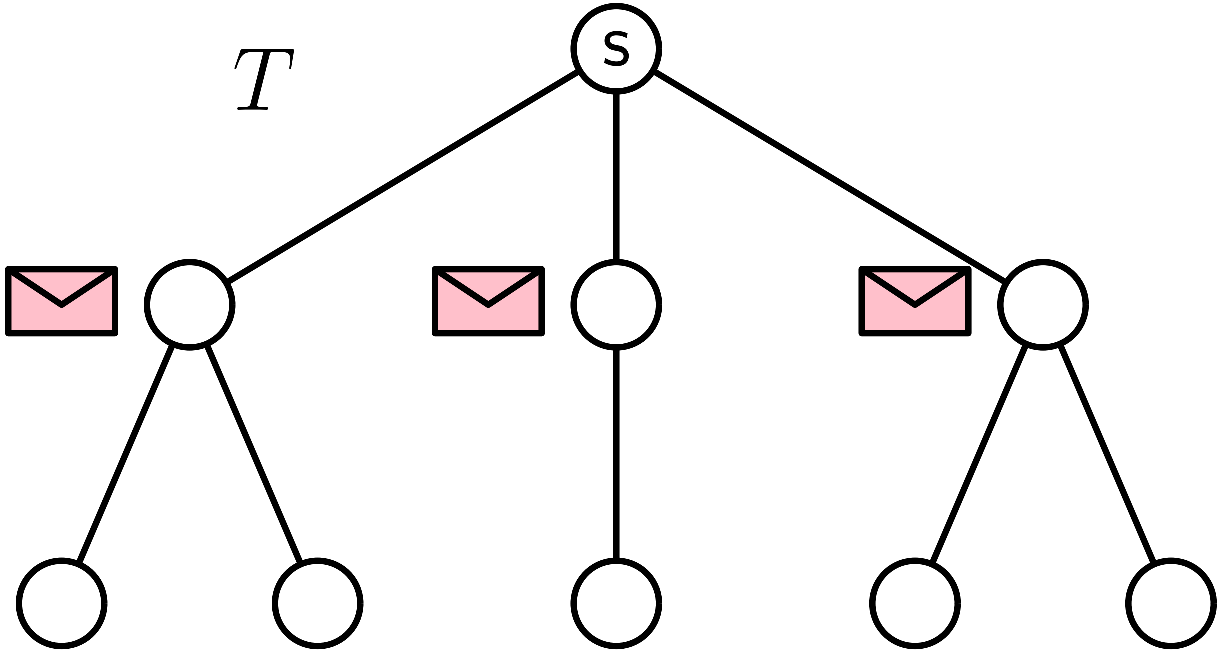
$$v_e(t_e, T) = \begin{cases} t_e & \text{if } e \in E(T) \\ 0 & \text{if } e \notin E(T) \end{cases}$$

Note: v_e represents a **cost** incurred by agent e!

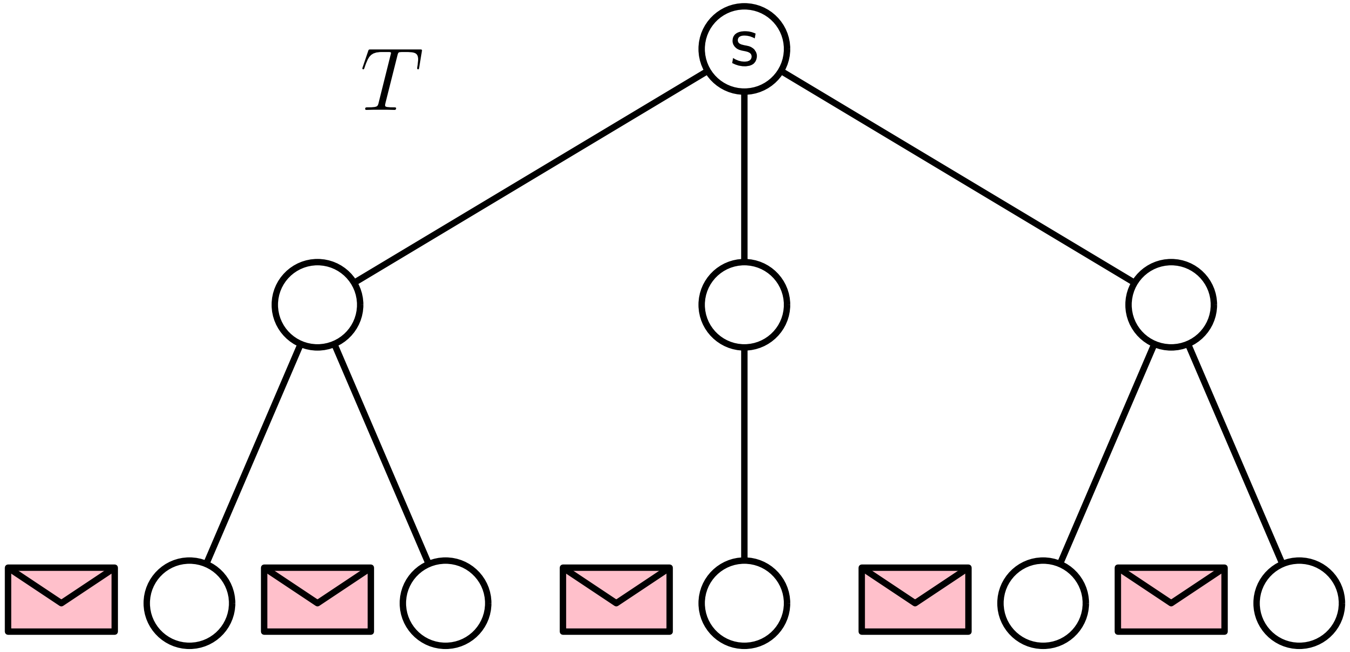
This models the multicast protocol



This models the multicast protocol

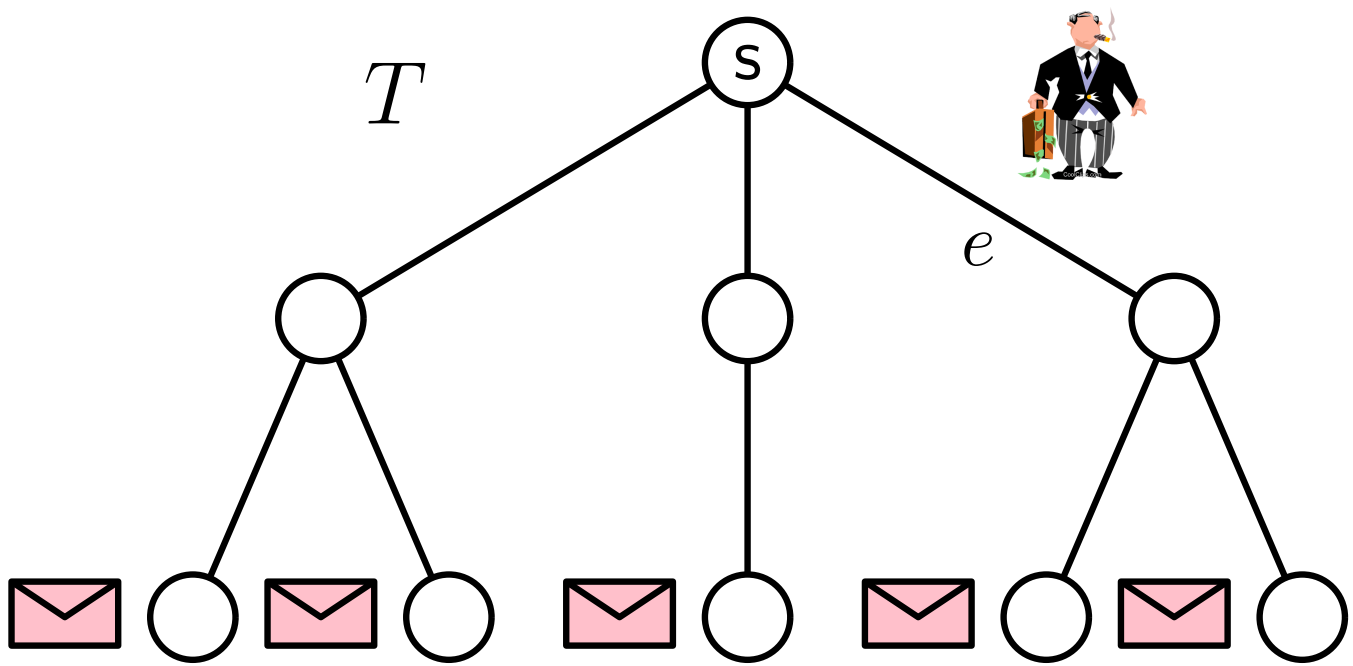


This models the multicast protocol



Each edge is traversed by a single (copy of the) message.

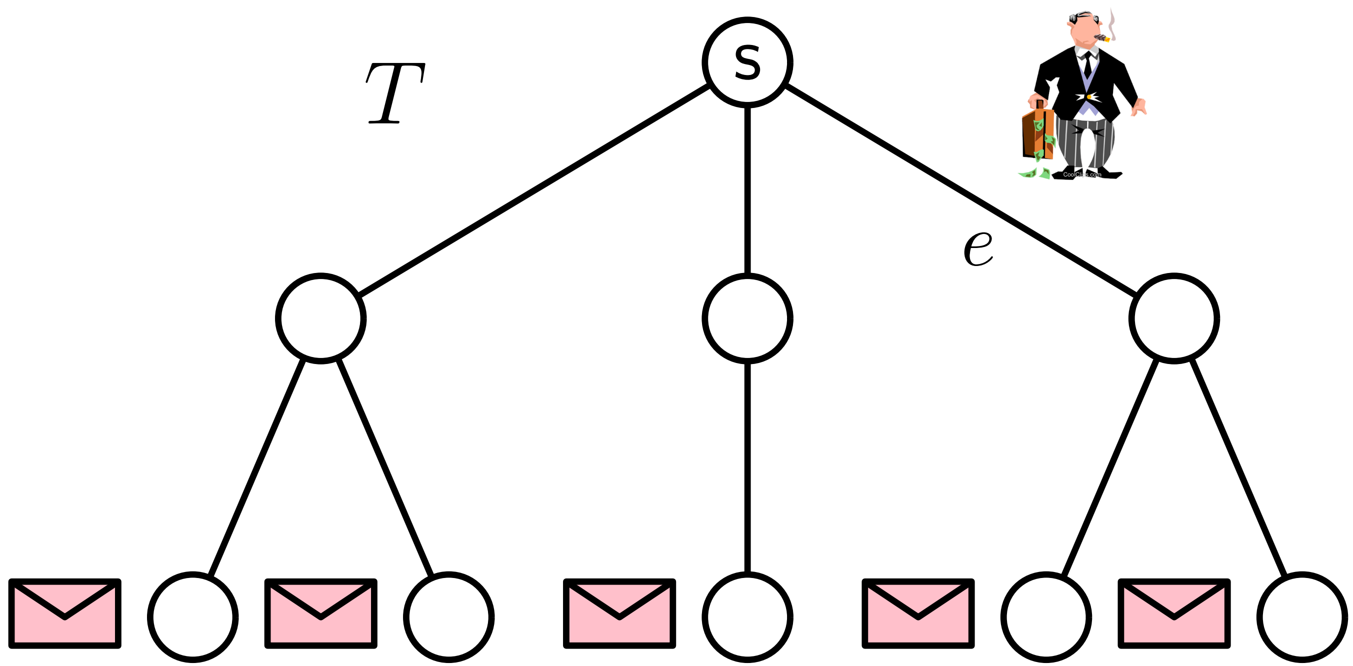
This models the multicast protocol



Each edge is traversed by a single (copy of the) message.

$$v_e(t_e, T) = \begin{cases} t_e & \text{if } e \in E(T) \\ 0 & \text{if } e \notin E(T) \end{cases}$$

This models the multicast protocol



Each edge is traversed by a single (copy of the) message.

$$v_e(t_e, T) = \begin{cases} t_e & \text{if } e \in E(T) \\ 0 & \text{if } e \notin E(T) \end{cases}$$

We want to minimize the time needed to deliver the message from s to each node: T must be a SPT.

The private-edge Shortest-Paths Tree (SPT) problem

Goal:

- Let $\mathcal{F}$ be the set of all spanning trees of G rooted at s
- We want to design a truthful mechanism that **minimizes** the following quantity w.r.t $T \in \mathcal{F}$:

$$f(t,T) = \sum_{v \in V} d_T(s,v) = \sum_{e \in E(T)} t_e \cdot \|e\|,$$

where $d_T(s,v)$ is the distance between s and v in T and $\|e\|$ is the number of source–node paths in T containing e.

The private-edge Shortest-Paths Tree (SPT) problem

Goal:

- Let $\mathcal{F}$ be the set of all spanning trees of G rooted at s
- We want to design a truthful mechanism that **minimizes** the following quantity w.r.t $T \in \mathcal{F}$:

$$f(t, T) = \sum_{v \in V} d_T(s, v) = \sum_{e \in E(T)} t_e \cdot \|e\|,$$

where $d_T(s, v)$ is the distance between s and v in T and $\|e\|$ is the number of source–node paths in T containing e.

Note:

$$f(t, T) = \sum_{e \in E(T)} t_e \cdot \|e\| \neq \sum_{e \in E(T)} t_e$$

The private-edge Shortest-Paths Tree (SPT) problem

Goal:

- Let $\mathcal{F}$ be the set of all spanning trees of G rooted at s
- We want to design a truthful mechanism that **minimizes** the following quantity w.r.t $T \in \mathcal{F}$:

$$f(t,T) = \sum_{v \in V} d_T(s,v) = \sum_{e \in E(T)} t_e \cdot \|e\|,$$

where $d_T(s,v)$ is the distance between s and v in T and $\|e\|$ is the number of source–node paths in T containing e.

Note:

$$f(t,T) = \sum_{e \in E(T)} t_e \cdot \|e\| \neq \sum_{e \in E(T)} t_e = \sum_{e \in E} v_e(t_e, T)$$

The private-edge Shortest-Paths Tree (SPT) problem

Goal:

- Let $\mathcal{F}$ be the set of all spanning trees of G rooted at s
- We want to design a truthful mechanism that **minimizes** the following quantity w.r.t $T \in \mathcal{F}$:

$$f(t,T) = \sum_{v \in V} d_T(s,v) = \sum_{e \in E(T)} t_e \cdot \|e\|,$$

where $d_T(s,v)$ is the distance between s and v in T and $\|e\|$ is the number of source–node paths in T containing e.

Note: **Non-utilitarian problem!**

$$f(t,T) = \sum_{e \in E(T)} t_e \cdot \|e\| \neq \sum_{e \in E(T)} t_e = \sum_{e \in E} v_e(t_e, T)$$

One-parameter Mechanism Design Problems

A mechanism design problem is *one-parameter* if:

- The private type of each player i is a *single parameter* $t_i \in \mathbb{R}$.

One-parameter Mechanism Design Problems

A mechanism design problem is *one-parameter* if:

- The private type of each player i is a *single parameter* $t_i \in \mathbb{R}$.

- The valuation function of player i w.r.t. an outcome o is of the form:
$$v_i(t_i, o) = t_i \cdot w_i(o),$$
where $w_i(o) \in \mathbb{R}_0^+$ is the *workload function* for agent i.

Is the private-edge SPT problem one-parameter?

- The private type of each player i is a *single parameter* $t_i \in \mathbb{R}$.

Is the private-edge SPT problem one-parameter?

- The private type of each player i is a *single parameter* $t_i \in \mathbb{R}$.

The type owned by each player is a single (positive) real number

Is the private-edge SPT problem one-parameter?

- The valuation function of player i w.r.t. an outcome o is of the form:

$$v_i(t_i, o) = t_i \cdot w_i(o),$$

where $w_i(o) \in \mathbb{R}_0^+$ is the *workload function* for agent i.

Is the private-edge SPT problem one-parameter?

- The valuation function of player i w.r.t. an outcome o is of the form:

$$v_i(t_i, o) = t_i \cdot w_i(o),$$

where $w_i(o) \in \mathbb{R}_0^+$ is the *workload function* for agent i.

$$v_e(t_e, T) = \begin{cases} t_e & \text{if } e \in E(T) \\ 0 & \text{if } e \notin E(T) \end{cases} \qquad = t_e \cdot w_e(T)$$

Is the private-edge SPT problem one-parameter?

- The valuation function of player i w.r.t. an outcome o is of the form:

$$v_i(t_i, o) = t_i \cdot w_i(o),$$

where $w_i(o) \in \mathbb{R}_0^+$ is the *workload function* for agent i.

$$v_e(t_e, T) = \begin{cases} t_e & \text{if } e \in E(T) \\ 0 & \text{if } e \notin E(T) \end{cases} \qquad = t_e \cdot w_e(T)$$

where

$$w_e(T) = \begin{cases} 1 & \text{if } e \in E(T) \\ 0 & \text{if } e \notin E(T) \end{cases}$$

The private-edge SPT problem is one-parameter!

- The private type of each player i is a *single parameter* $t_i \in \mathbb{R}$.

- The valuation function of player i w.r.t. an outcome o is of the form:

$$v_i(t_i, o) = t_i \cdot w_i(o),$$

where $w_i(o) \in \mathbb{R}_0^+$ is the *workload function* for agent i.

The private-edge SPT problem is one-parameter!

- The private type of each player i is a *single parameter* $t_i \in \mathbb{R}$.

- The valuation function of player i w.r.t. an outcome o is of the form:

$$v_i(t_i, o) = t_i \cdot w_i(o),$$

where $w_i(o) \in \mathbb{R}_0^+$ is the *workload function* for agent i.

$$\Downarrow$$

The private-edge SPT problem is one-parameter!

A necessary condition for designing OP truthful mechanisms

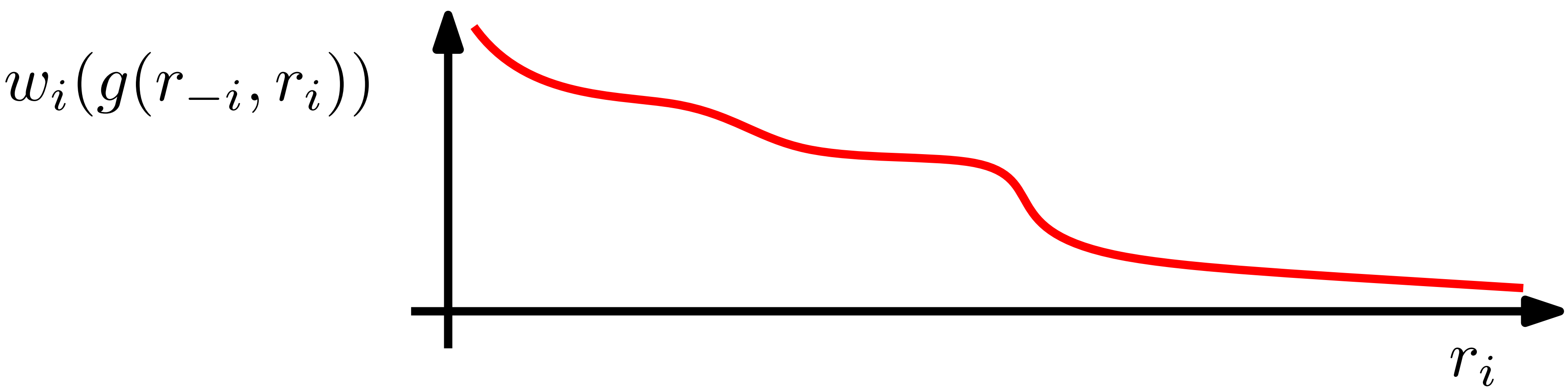
Definition: An algorithm g for a minimization OP problem is *monotone* if, $\forall$ player i, and $\forall r_{-i} = (r_1, \ldots, r_{i-1}, r_{i+1}, r_N)$ it holds that:

$$w_i(g(r_{-i}, r_i)) \text{ is non-increasing w.r.t. } r_i.$$

A necessary condition for designing OP truthful mechanisms

Definition: An algorithm g for a minimization OP problem is *monotone* if, $\forall$ player i, and $\forall r_{-i} = (r_1, \ldots, r_{i-1}, r_{i+1}, r_N)$ it holds that:

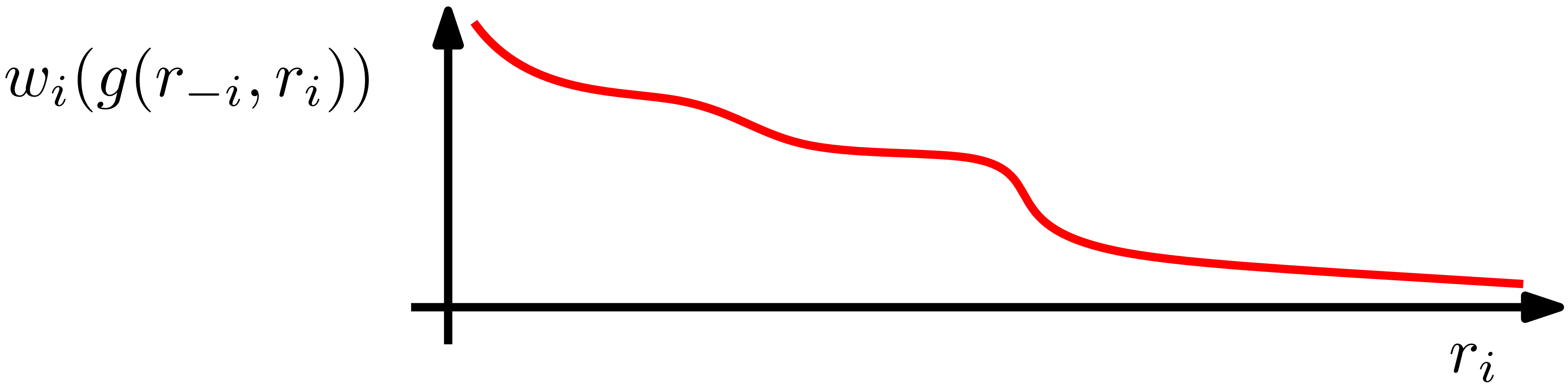
$$w_i(g(r_{-i}, r_i)) \text{ is non-increasing w.r.t. } r_i.$$



A necessary condition for designing OP truthful mechanisms

Definition: An algorithm g for a minimization OP problem is *monotone* if, $\forall$ player i, and $\forall r_{-i} = (r_1, \ldots, r_{i-1}, r_{i+1}, r_N)$ it holds that:

$$w_i(g(r_{-i}, r_i)) \text{ is non-increasing w.r.t. } r_i.$$



Theorem (R.B. Myerson, 1981):
A mechanism $M = \langle g, p \rangle$ for a minimization OP problem is truthful **only if** g is monotone.

Theorem (R.B. Myerson, 1981):
A mechanism $M = \langle g, p \rangle$ for a minimization OP problem is truthful **only if** g is monotone.

Proof (by contradiction):

Theorem (R.B. Myerson, 1981):
A mechanism $M = \langle g, p \rangle$ for a minimization OP problem is truthful **only if** g is monotone.

Proof (by contradiction):

- Assume that there exists a truthful mechanism $M = \langle g, p \rangle$ such that g is non-monotone.

Theorem (R.B. Myerson, 1981):
A mechanism $M = \langle g, p \rangle$ for a minimization OP problem is truthful **only if** g is monotone.

Proof (by contradiction):

- Assume that there exists a truthful mechanism $M = \langle g, p \rangle$ such that g is non-monotone.

- There is a player i and a vector r_{-i} of strategies such that $w_i(g(r_{-i}, r_i))$ is not monotonically non-increasing w.r.t. r_i.

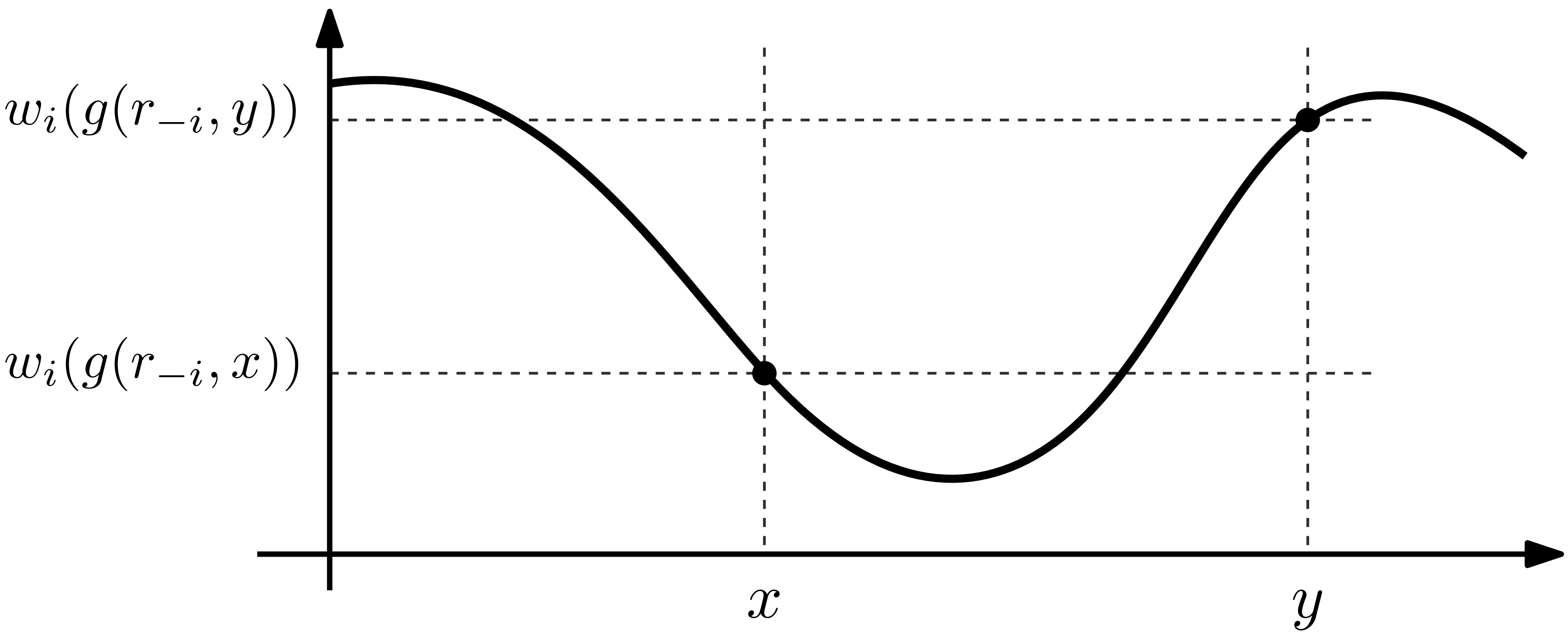
Theorem (R.B. Myerson, 1981):
A mechanism $M = \langle g, p \rangle$ for a minimization OP problem is truthful **only if** g is monotone.

Proof (by contradiction):

- Assume that there exists a truthful mechanism $M = \langle g, p \rangle$ such that g is non-monotone.
- There is a player i and a vector r_{-i} of strategies such that $w_i(g(r_{-i}, r_i))$ is not monotonically non-increasing w.r.t. r_i.
- There exists $x, y \in \mathbb{R}$ such that $x < y$ and $w_i(g(r_{-i}, x)) < w_i(g(r_{-i}, y))$

Proof (cont.):

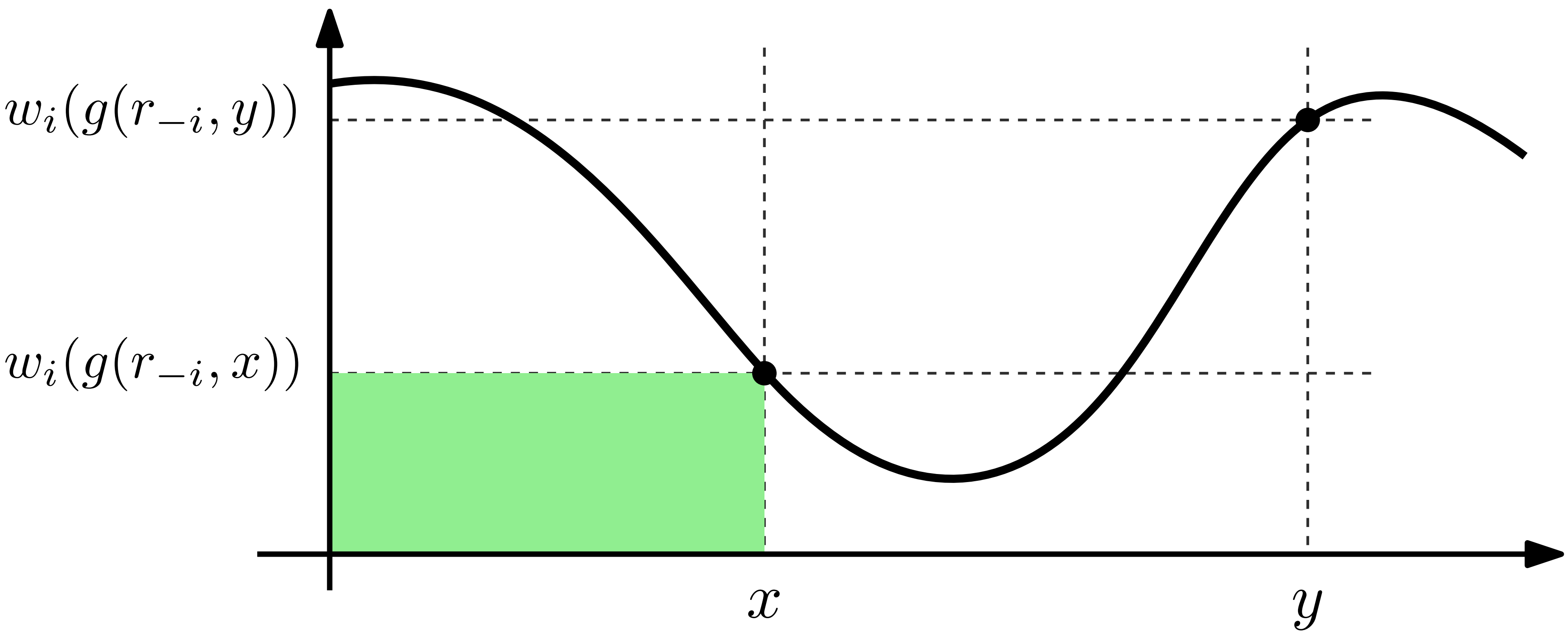
Consider $t_i = x$:



Proof (cont.):

Consider $t_i = x$:

If $r_i = x$, $v(t_i, o) = x \cdot w_i(g(r_{-i}, x))$

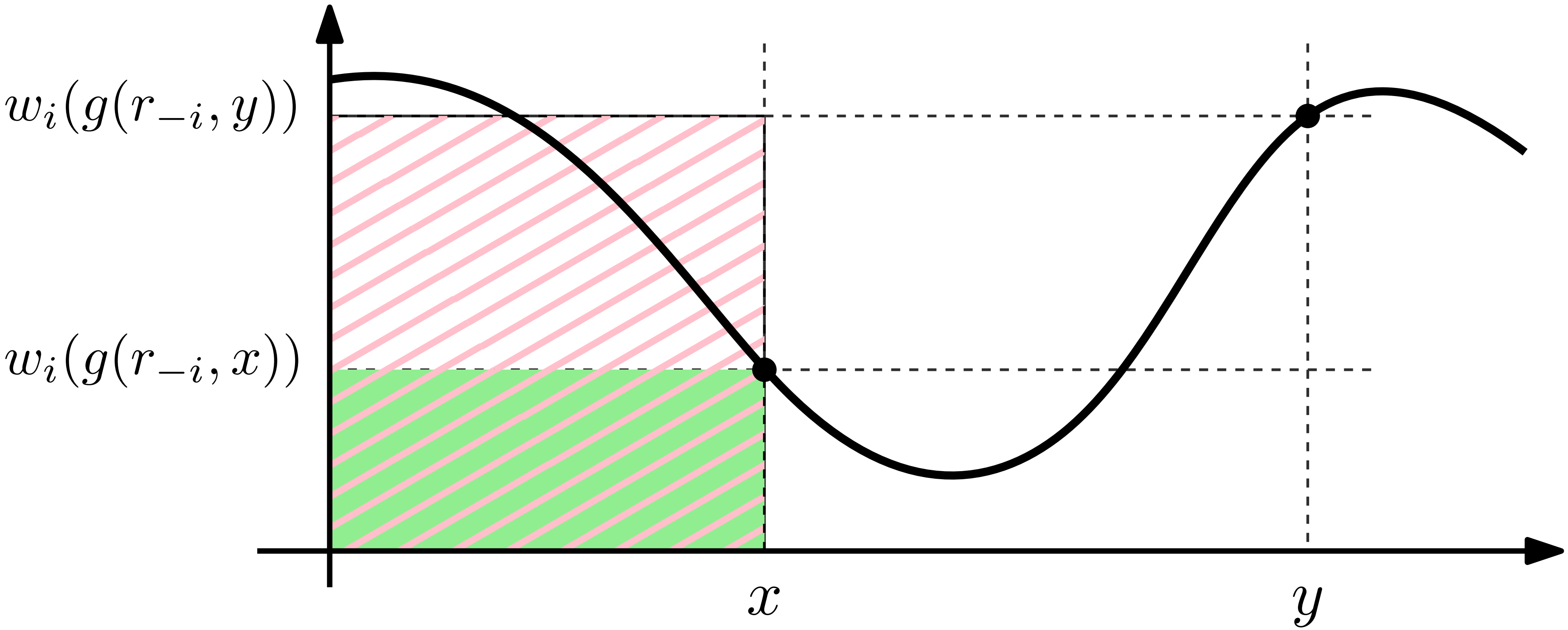


Proof (cont.):

Consider $t_i = x$:

If $r_i = x$, $v(t_i, o) = x \cdot w_i(g(r_{-i}, x))$

If $r_i = y$, $v(t_i, o) = x \cdot w_i(g(r_{-i}, y))$

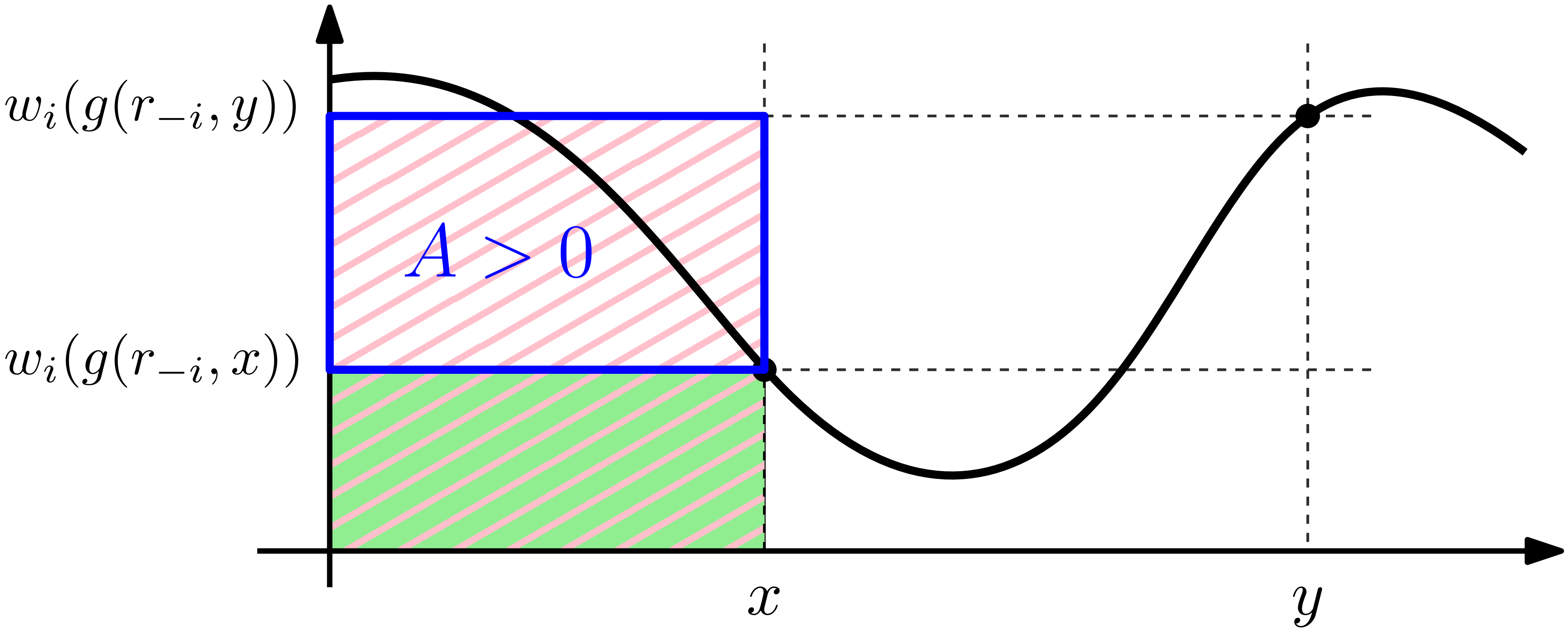


Proof (cont.):

Consider $t_i = x$:

If $r_i = x$, $v(t_i, o) = x \cdot w_i(g(r_{-i}, x))$

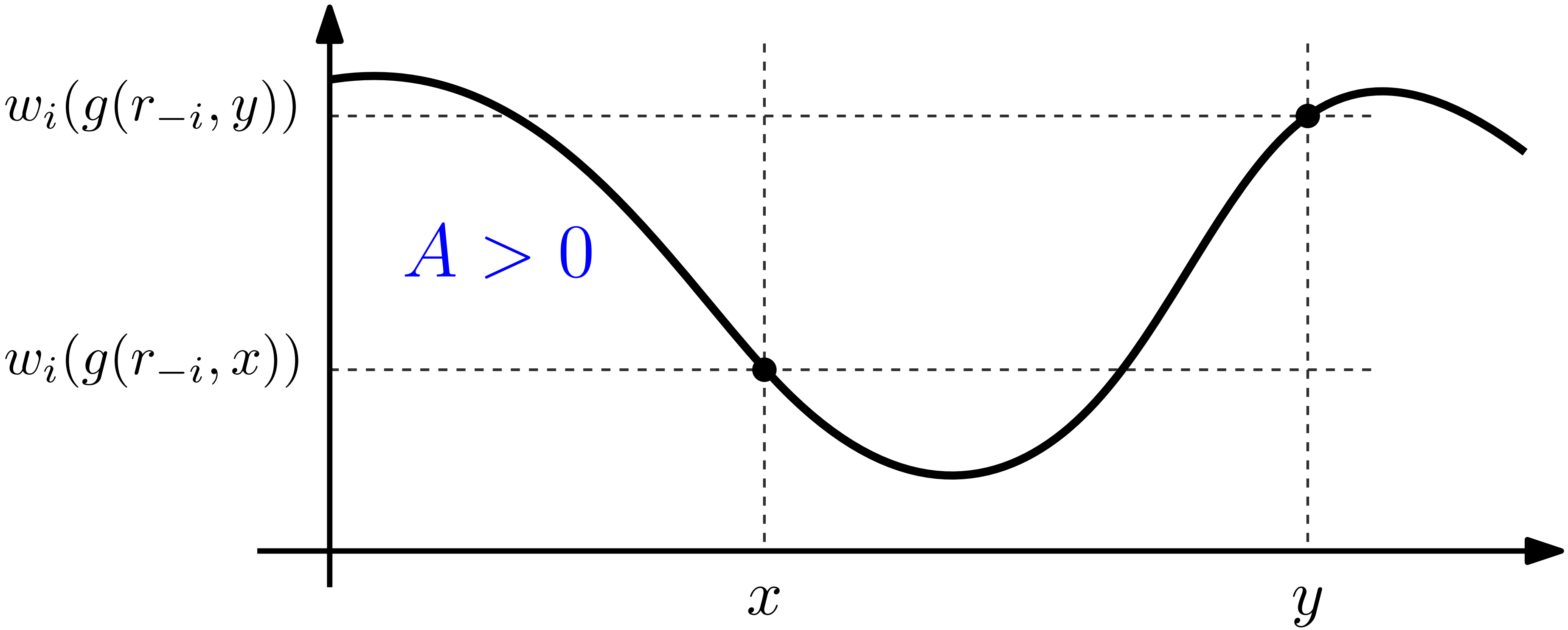
If $r_i = y$, $v(t_i, o) = x \cdot w_i(g(r_{-i}, y))$



If $t_i = x$, $v(t_i, \cdot)$ <u>increases</u> by A when reporting y instead of x.

Proof (cont.):

Consider $t_i = y$:

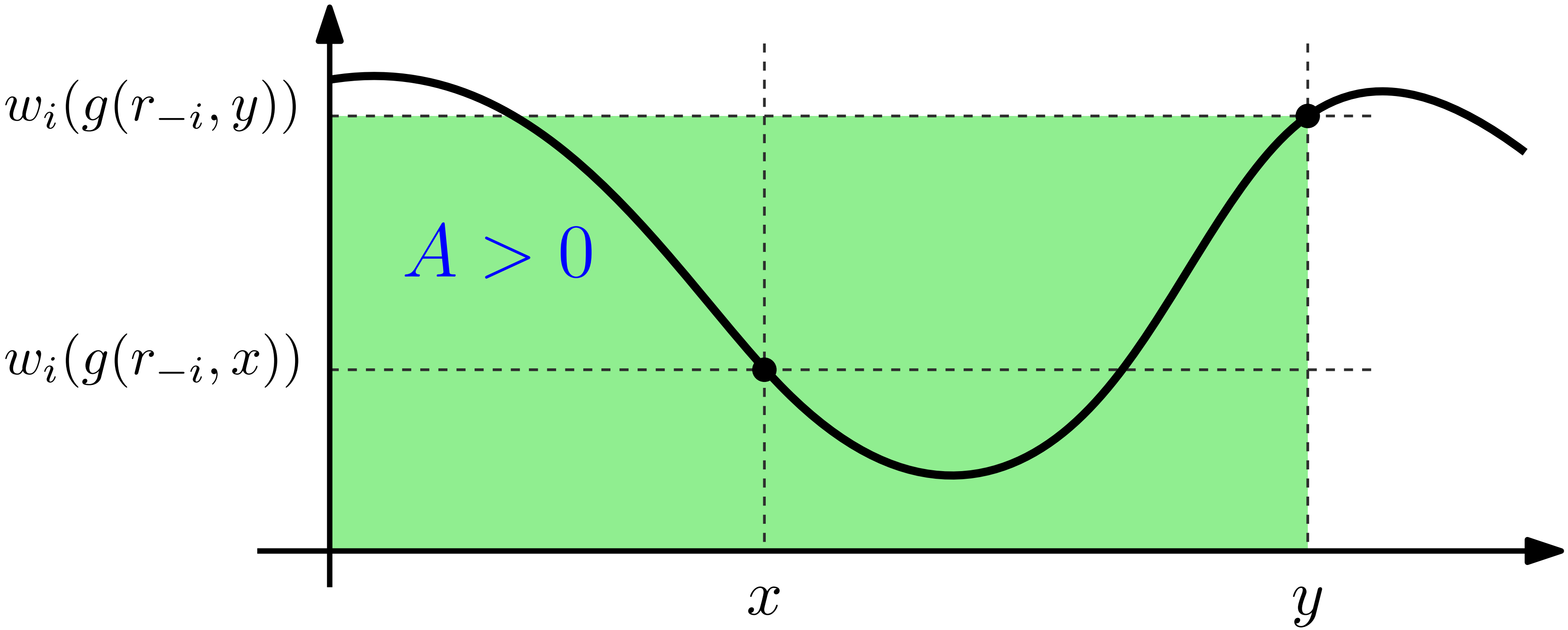


If $t_i = x$, $v(t_i, \cdot)$ increases by A when reporting y instead of x.

Proof (cont.):

Consider $t_i = y$:

If $r_i = y$, $v(t_i, o) = y \cdot w_i(g(r_{-i}, y))$



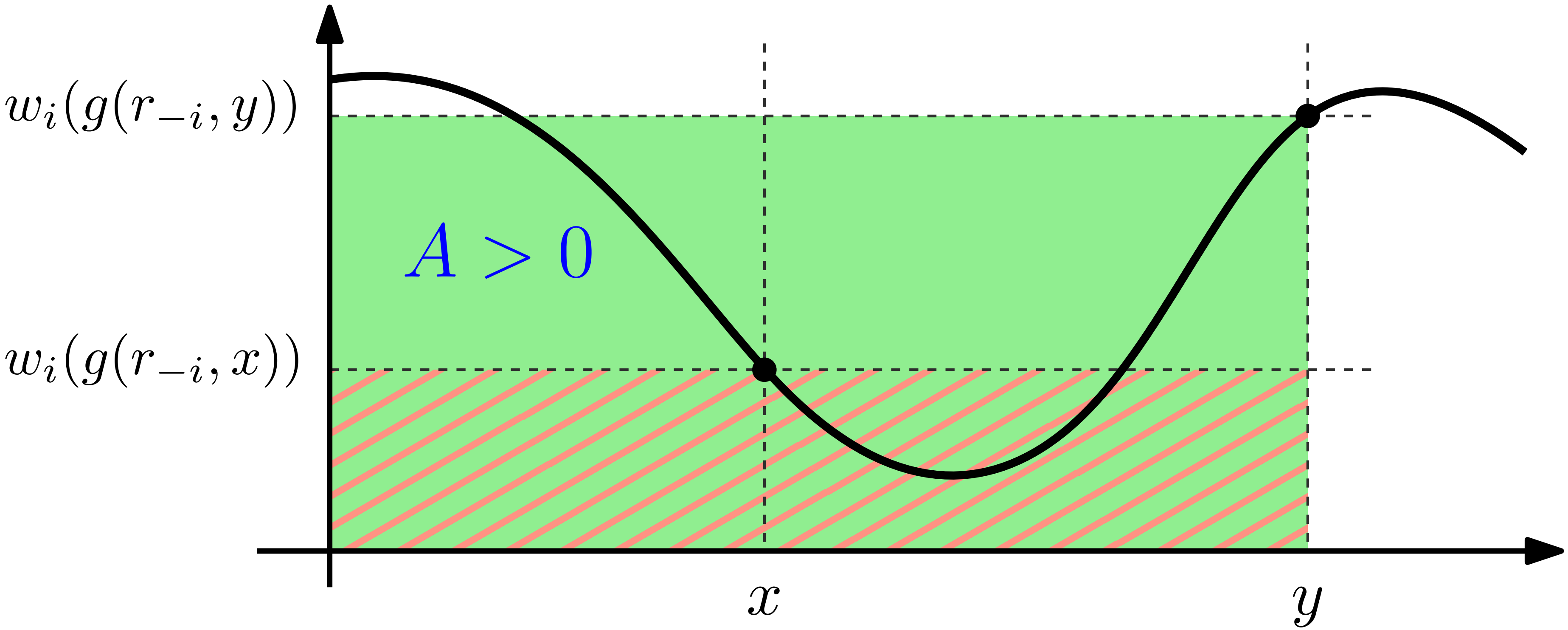
If $t_i = x$, $v(t_i, \cdot)$ <u>increases</u> by A when reporting y instead of x.

Proof (cont.):

Consider $t_i = y$:

If $r_i = y$, $v(t_i, o) = y \cdot w_i(g(r_{-i}, y))$

If $r_i = x$, $v(t_i, o) = y \cdot w_i(g(r_{-i}, x))$



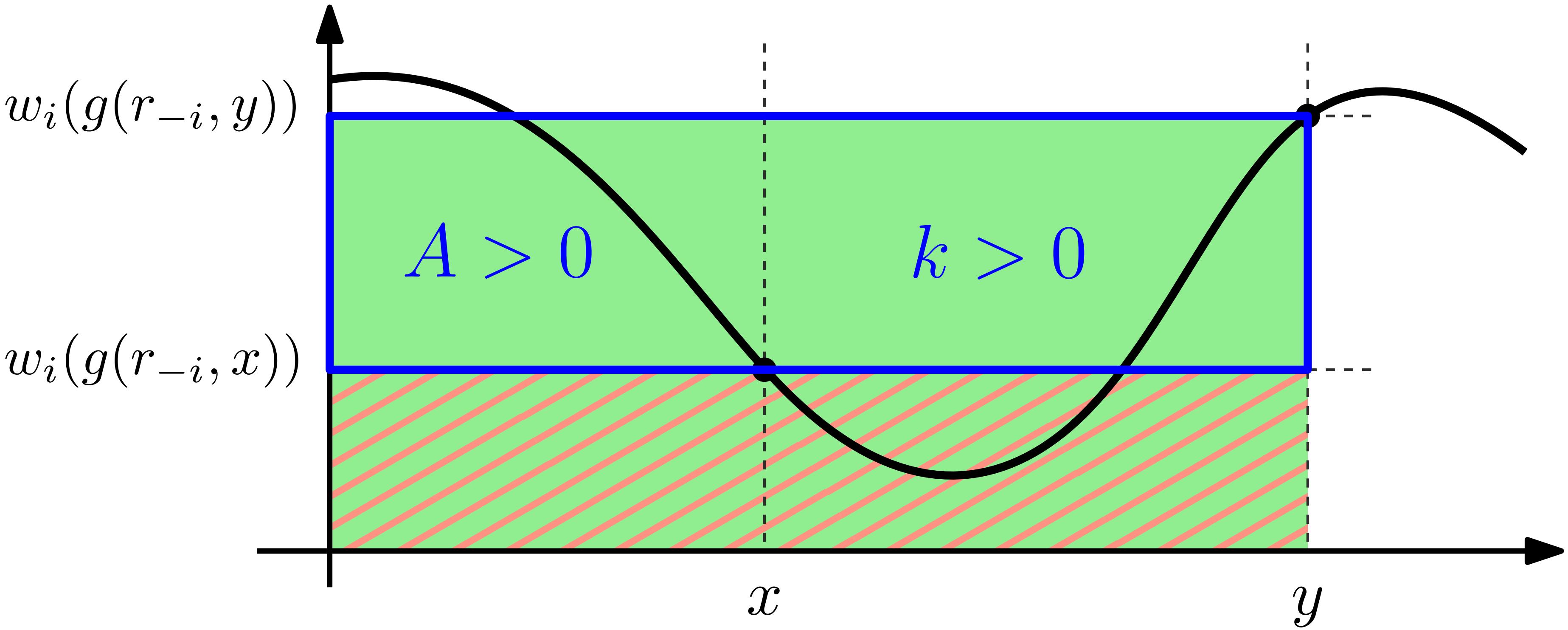
If $t_i = x$, $v(t_i, \cdot)$ increases by A when reporting y instead of x.

Proof (cont.):

Consider $t_i = y$:

If $r_i = y$, $v(t_i, o) = y \cdot w_i(g(r_{-i}, y))$

If $r_i = x$, $v(t_i, o) = y \cdot w_i(g(r_{-i}, x))$



If $t_i = x$, $v(t_i, \cdot)$ increases by A when reporting y instead of x.

If $t_i = y$, $v(t_i, \cdot)$ decreases by $A+k$ when reporting x instead of y.

Proof (cont.):

If $t_i = x$, $v(t_i, \cdot)$ increases by A when reporting y instead of x.

If $t_i = y$, $v(t_i, \cdot)$ decreases by $A + k$ when reporting x instead of y.

Let $\Delta p = p_i(r_{-i}, y) - p_i(r_{-i}, x)$ be the difference in the payment received by player i when she reports y instead of x.

Proof (cont.):

If $t_i = x$, $v(t_i, \cdot)$ <u>increases</u> by A when reporting y instead of x.

If $t_i = y$, $v(t_i, \cdot)$ <u>decreases</u> by $A+k$ when reporting x instead of y.

Let $\Delta p = p_i(r_{-i}, y) - p_i(r_{-i}, x)$ be the difference in the payment received by player i when she reports y instead of x.

If $\Delta p > A$ then player i has an incentive to lie when $t_i = x$.

(report y: cost increases by A, payment increases by $\Delta p > A$)

Proof (cont.):

If $t_i = x$, $v(t_i, \cdot)$ <u>increases</u> by A when reporting y instead of x.

If $t_i = y$, $v(t_i, \cdot)$ <u>decreases</u> by $A+k$ when reporting x instead of y.

Let $\Delta p = p_i(r_{-i}, y) - p_i(r_{-i}, x)$ be the difference in the payment received by player i when she reports y instead of x.

We must have $\Delta p \leq A$

Proof (cont.):

If $t_i = x$, $v(t_i, \cdot)$ <u>increases</u> by A when reporting y instead of x.

If $t_i = y$, $v(t_i, \cdot)$ <u>decreases</u> by $A+k$ when reporting x instead of y.

Let $\Delta p = p_i(r_{-i}, y) - p_i(r_{-i}, x)$ be the difference in the payment received by player i when she reports y instead of x.

We must have $\Delta p \leq A$

If $\Delta p < A + k$ then player i has an incentive to lie when $t_i = y$.

(report y: cost decreases by $A + k$, payment decreases by $\Delta p < A + k$)

Proof (cont.):

If $t_i = x$, $v(t_i, \cdot)$ increases by A when reporting y instead of x.

If $t_i = y$, $v(t_i, \cdot)$ decreases by $A+k$ when reporting x instead of y.

Let $\Delta p = p_i(r_{-i}, y) - p_i(r_{-i}, x)$ be the difference in the payment received by player i when she reports y instead of x.

We must have $\Delta p \leq A$

But simultaneously $\Delta p \geq A + k > A$ (since $k > 0$)

Proof (cont.):

If $t_i = x$, $v(t_i, \cdot)$ <u>increases</u> by A when reporting y instead of x.

If $t_i = y$, $v(t_i, \cdot)$ <u>decreases</u> by $A+k$ when reporting x instead of y.

Let $\Delta p = p_i(r_{-i}, y) - p_i(r_{-i}, x)$ be the difference in the payment received by player i when she reports y instead of x.

We must have $\Delta p \leq A$

But simultaneously $\Delta p \geq A + k > A$ (since $k > 0$)

$\square$

One-parameter Mechanisms

A one-parameter (OP) mechanism (for a OP problem) is a pair $M = \langle g, p \rangle$ such that:

- g is any monotone algorithm (for the underlying OP problem)

- $p_i(r) = h_i(r_{-i}) + r_i w_i(r) - \int_0^{r_i} w_i(r_{-i}, z) dz$

where $h_i(r_{-i})$ is an arbitrary function *independent of* r_i.

One-parameter Mechanisms

A one-parameter (OP) mechanism (for a OP problem) is a pair $M = \langle g, p \rangle$ such that:

- g is any monotone algorithm (for the underlying OP problem)

- $p_i(r) = h_i(r_{-i}) + \underline{r_i w_i(r)} - \int_0^{r_i} w_i(r_{-i}, z)dz$

where $h_i(r_{-i})$ is an arbitrary function *independent of* r_i.

To simplify notation we will write $w_i(r)$ in place of $w_i(g(r))$.

One-parameter Mechanisms

Theorem (R.B. Myerson, 1981):
An one-parameter mechanism (for an OP problem) is truthful.

One-parameter Mechanisms

Theorem (R.B. Myerson, 1981):
An one-parameter mechanism (for an OP problem) is truthful.

Proof:

- We show that the utility of player i can only decrease when she lies.

$$p_i(r) = h_i(r_{-i}) + r_i w_i(r) - \int_0^{r_i} w_i(r_{-i}, z)dz$$

One-parameter Mechanisms

Theorem (R.B. Myerson, 1981):
An one-parameter mechanism (for an OP problem) is truthful.

Proof:

- We show that the utility of player i can only decrease when she lies.

$$p_i(r) = \underbrace{h_i(r_{-i})}_{\substack{\text{does not depend on } r_i \\ \text{and can be set to } 0}} + r_i w_i(r) - \int_0^{r_i} w_i(r_{-i}, z) dz$$

One-parameter Mechanisms

Theorem (R.B. Myerson, 1981):
An one-parameter mechanism (for an OP problem) is truthful.

Proof:

- We show that the utility of player i can only decrease when she lies.

$$p_i(r) = r_i w_i(r) - \int_0^{r_i} w_i(r_{-i}, z)dz$$

(This will produce negative utilities)

Proof (cont.):

- When $r_i = t_i$:

$$u_i(t_i, g(r_{-i}, t_i)) = p_i(r_{-i}, t_i) - v_i(t_i, g(r_{-i}, t_i))$$

Proof (cont.):

- When $r_i = t_i$:

$$u_i(t_i, g(r_{-i}, t_i)) = t_i w_i(r_{-i}, t_i) - \int_0^{r_i} w_i(r_{-i}, z)dz - t_i w_i(r_{-i}, t_i)$$

Proof (cont.):

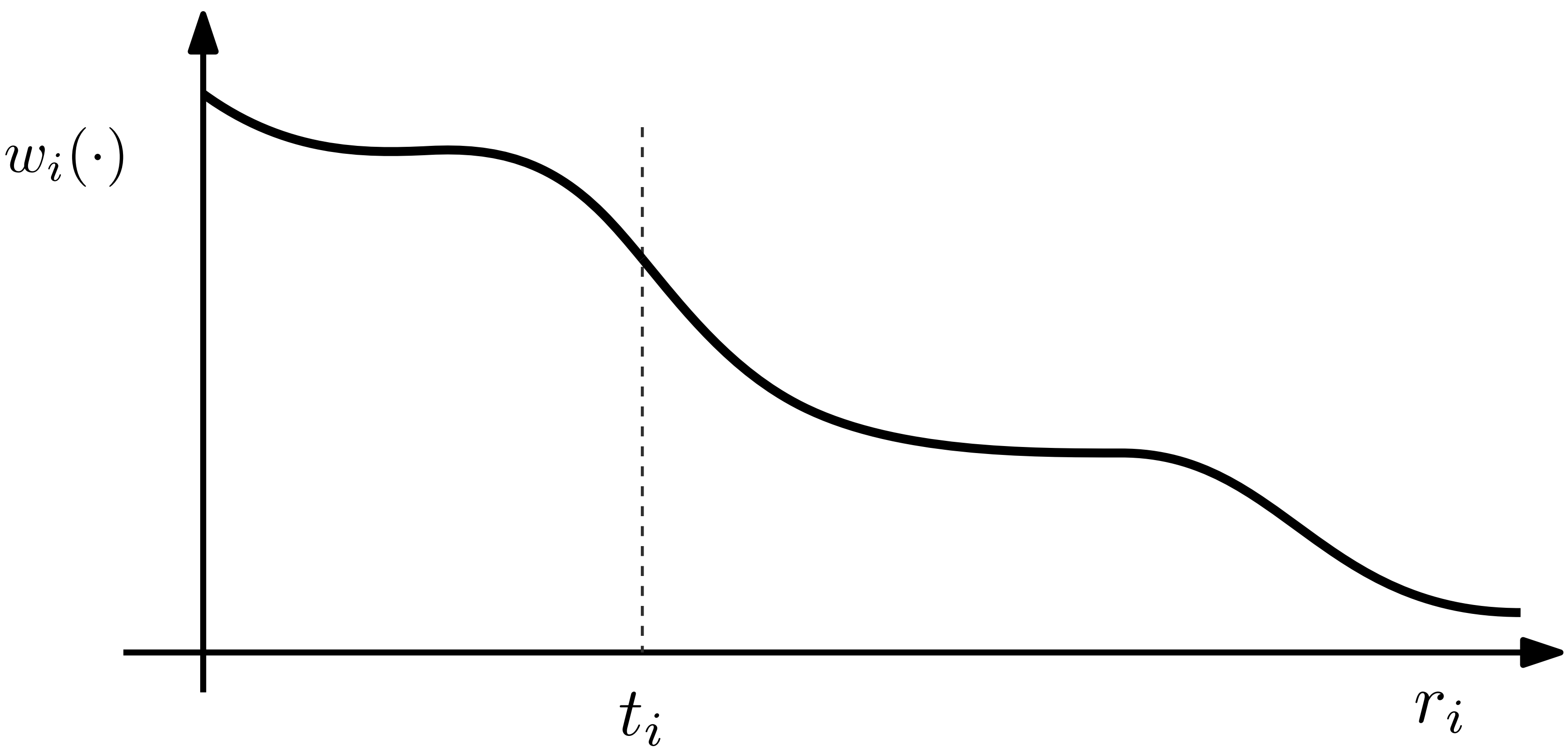
- When $r_i = t_i$:

$$u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$$

Proof (cont.):

- When $r_i = t_i$:

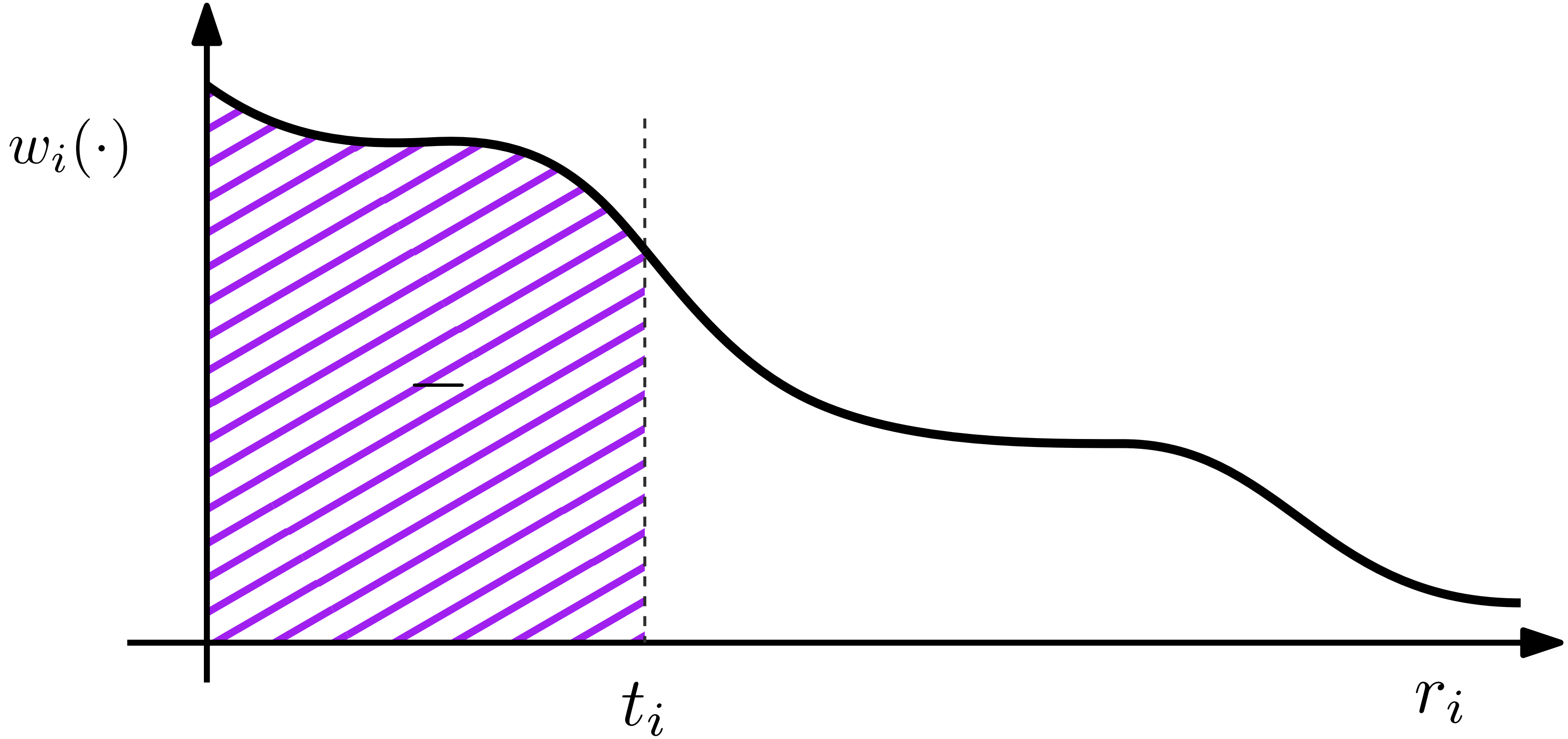
$$u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$$



Proof (cont.):

- When $r_i = t_i$:

$$u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$$



Proof (cont.):

- When $r_i = t_i$: $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$
- When $r_i > t_i$:

$$u_i(t_i, g(r_{-i}, t_i)) = p_i(r_{-i}, r_i) - v_i(t_i, g(r_{-i}, r_i))$$

Proof (cont.):

- When $r_i = t_i$: $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$
- When $r_i > t_i$:

$$u_i(t_i, g(r_{-i}, t_i)) = r_i w_i(r_{-i}, r_i) - \int_0^{r_i} w_i(r_{-i}, z)dz - t_i w_i(r_{-i}, r_i)$$

Proof (cont.):

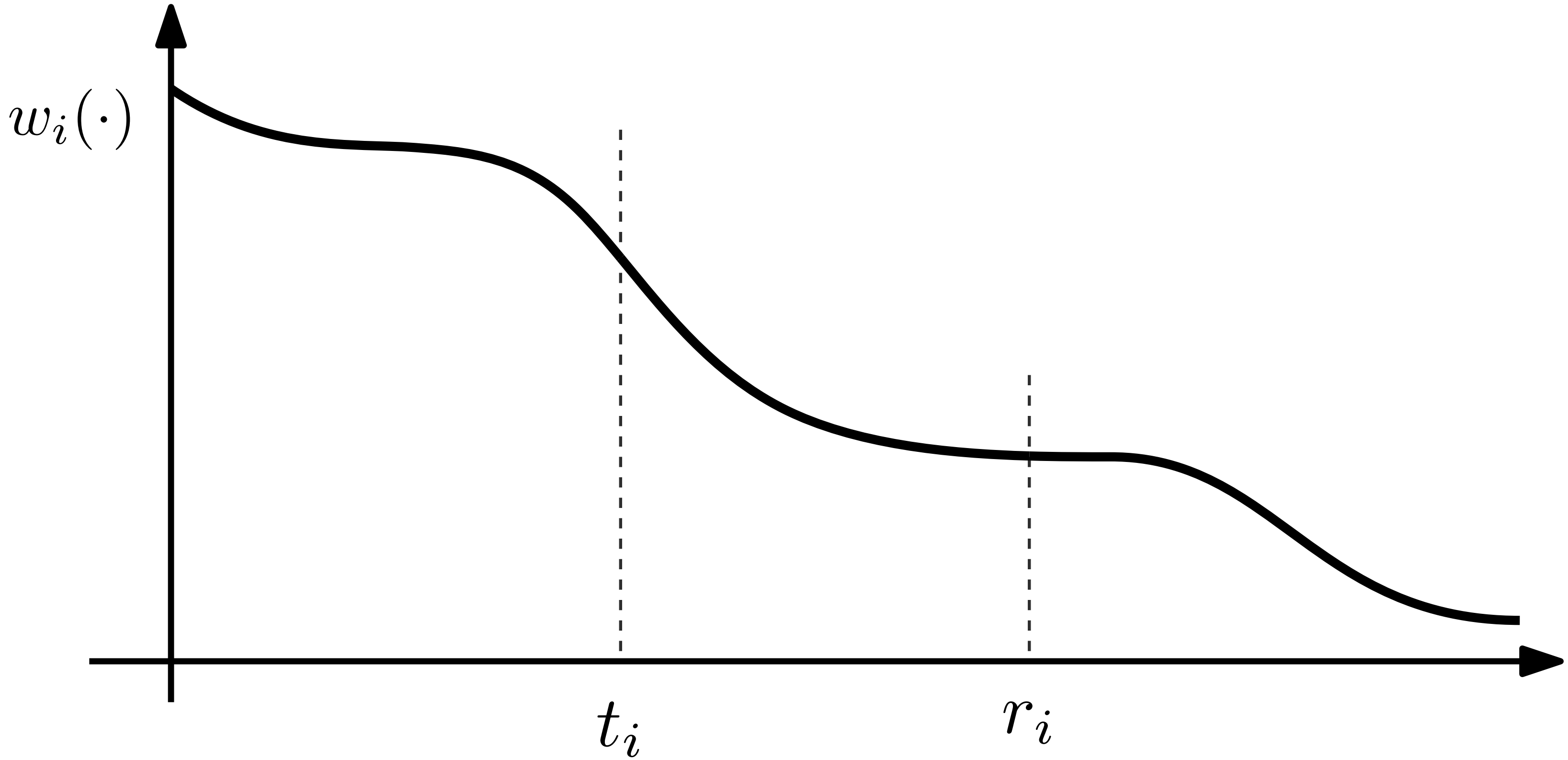
- When $r_i = t_i$: $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$
- When $r_i > t_i$:

$$u_i(t_i, g(r_{-i}, t_i)) = (r_i - t_i)w_i(r_{-i}, r_i) - \int_0^{r_i} w_i(r_{-i}, z)dz$$

Proof (cont.):

- When $r_i = t_i$: $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$
- When $r_i > t_i$:

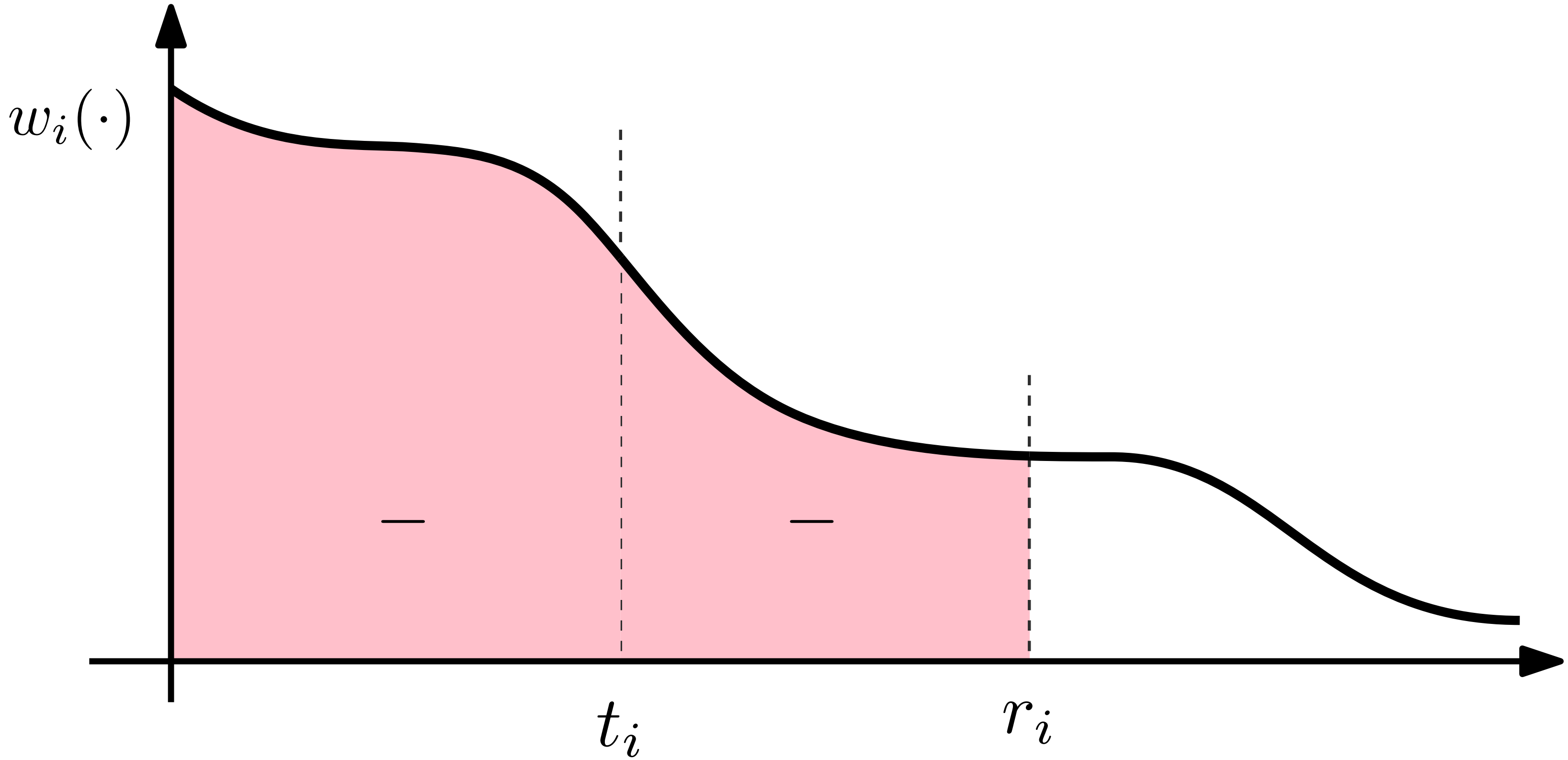
$$u_i(t_i, g(r_{-i}, t_i)) = (r_i - t_i)w_i(r_{-i}, r_i) - \int_0^{r_i} w_i(r_{-i}, z)dz$$



Proof (cont.):

- When $r_i = t_i$: $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$
- When $r_i > t_i$:

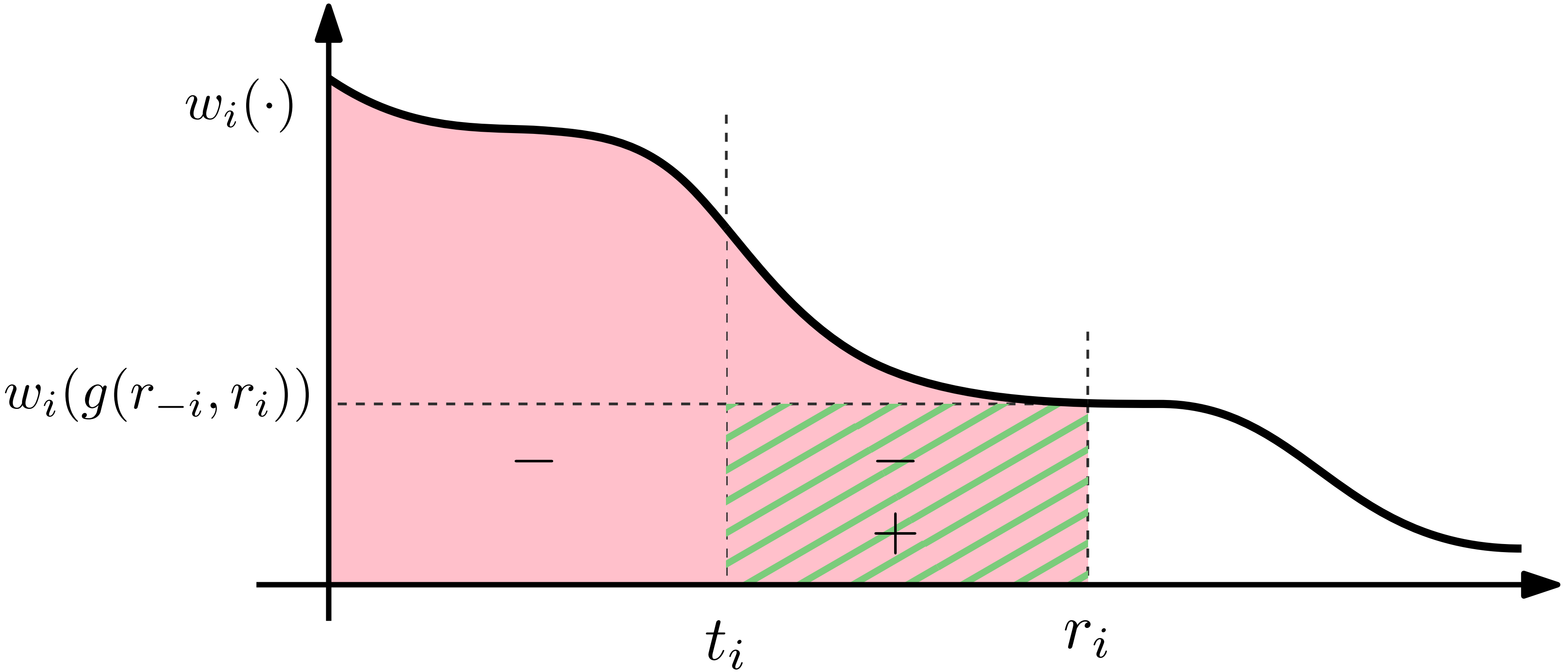
$$u_i(t_i, g(r_{-i}, t_i)) = (r_i - t_i)w_i(r_{-i}, r_i) - \underline{\int_0^{r_i} w_i(r_{-i}, z)dz}$$



Proof (cont.):

- When $r_i = t_i$: $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$
- When $r_i > t_i$:

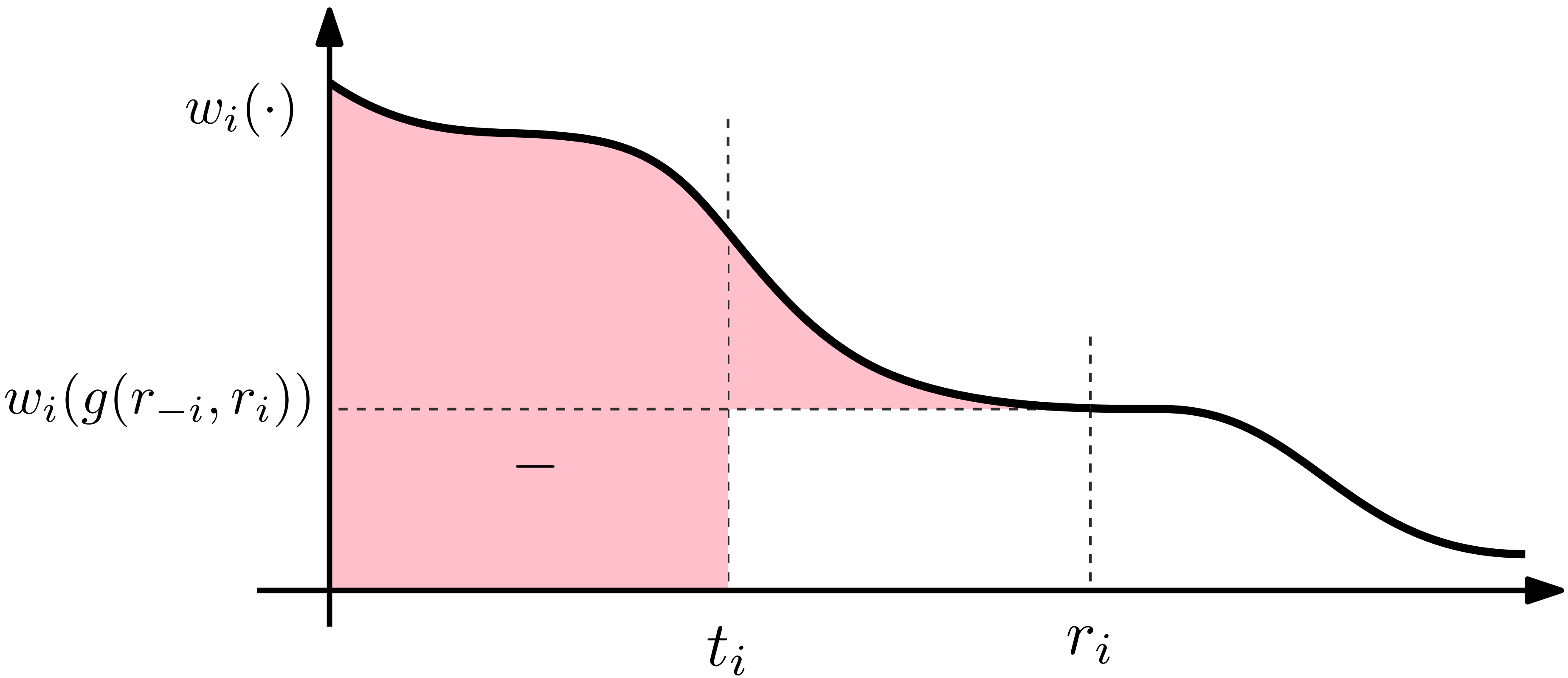
$$u_i(t_i, g(r_{-i}, t_i)) = (r_i - t_i)w_i(r_{-i}, r_i) - \int_0^{r_i} w_i(r_{-i}, z)dz$$



Proof (cont.):

- When $r_i = t_i$: $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$
- When $r_i > t_i$:

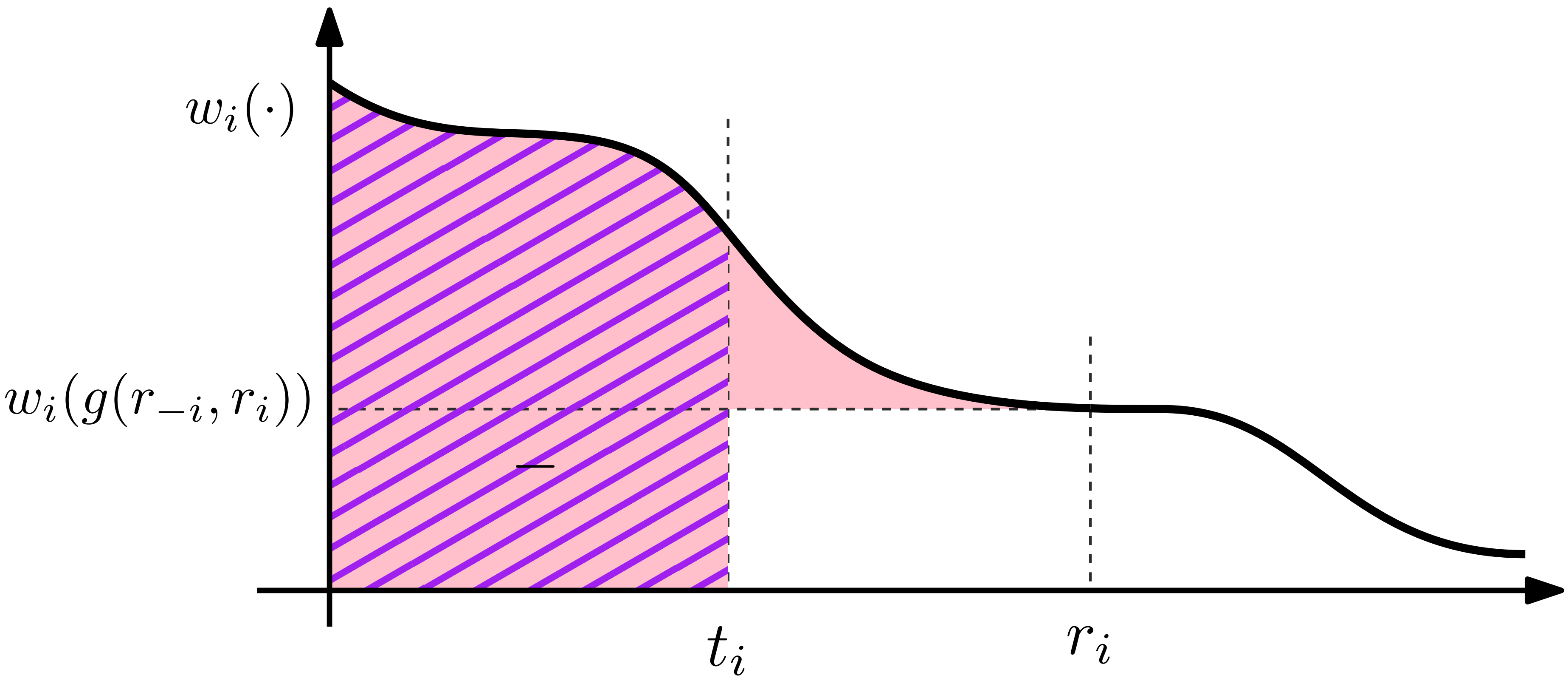
$$u_i(t_i, g(r_{-i}, t_i)) = (r_i - t_i)w_i(r_{-i}, r_i) - \int_0^{r_i} w_i(r_{-i}, z)dz$$



Proof (cont.):

- When $r_i = t_i$: $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$
- When $r_i > t_i$:

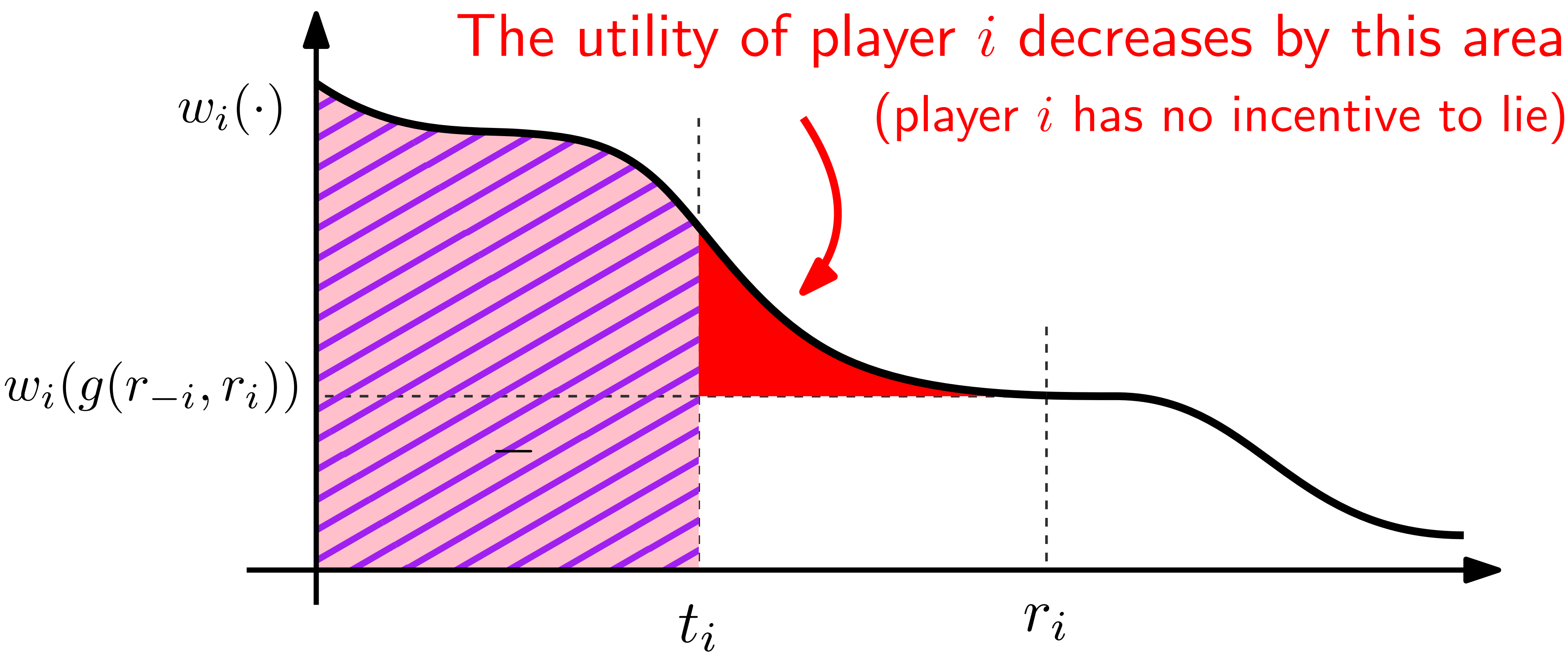
$$u_i(t_i, g(r_{-i}, t_i)) = (r_i - t_i)w_i(r_{-i}, r_i) - \int_0^{r_i} w_i(r_{-i}, z)dz$$



Proof (cont.):

- When $r_i = t_i$: $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$
- When $r_i > t_i$:

$$u_i(t_i, g(r_{-i}, t_i)) = (r_i - t_i)w_i(r_{-i}, r_i) - \int_0^{r_i} w_i(r_{-i}, z)dz$$



Proof (cont.):

- When $r_i = t_i$: $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$
- When $r_i < t_i$:

$$u_i(t_i, g(r_{-i}, r_i)) = p_i(r_{-i}, r_i) - v_i(t_i, g(r_{-i}, r_i))$$

Proof (cont.):

- When $r_i = t_i$: $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$
- When $r_i < t_i$:

$$u_i(t_i, g(r_{-i}, t_i)) = r_i w_i(r_{-i}, r_i) - \int_0^{r_i} w_i(r_{-i}, z)dz - t_i w_i(r_{-i}, r_i)$$

Proof (cont.):

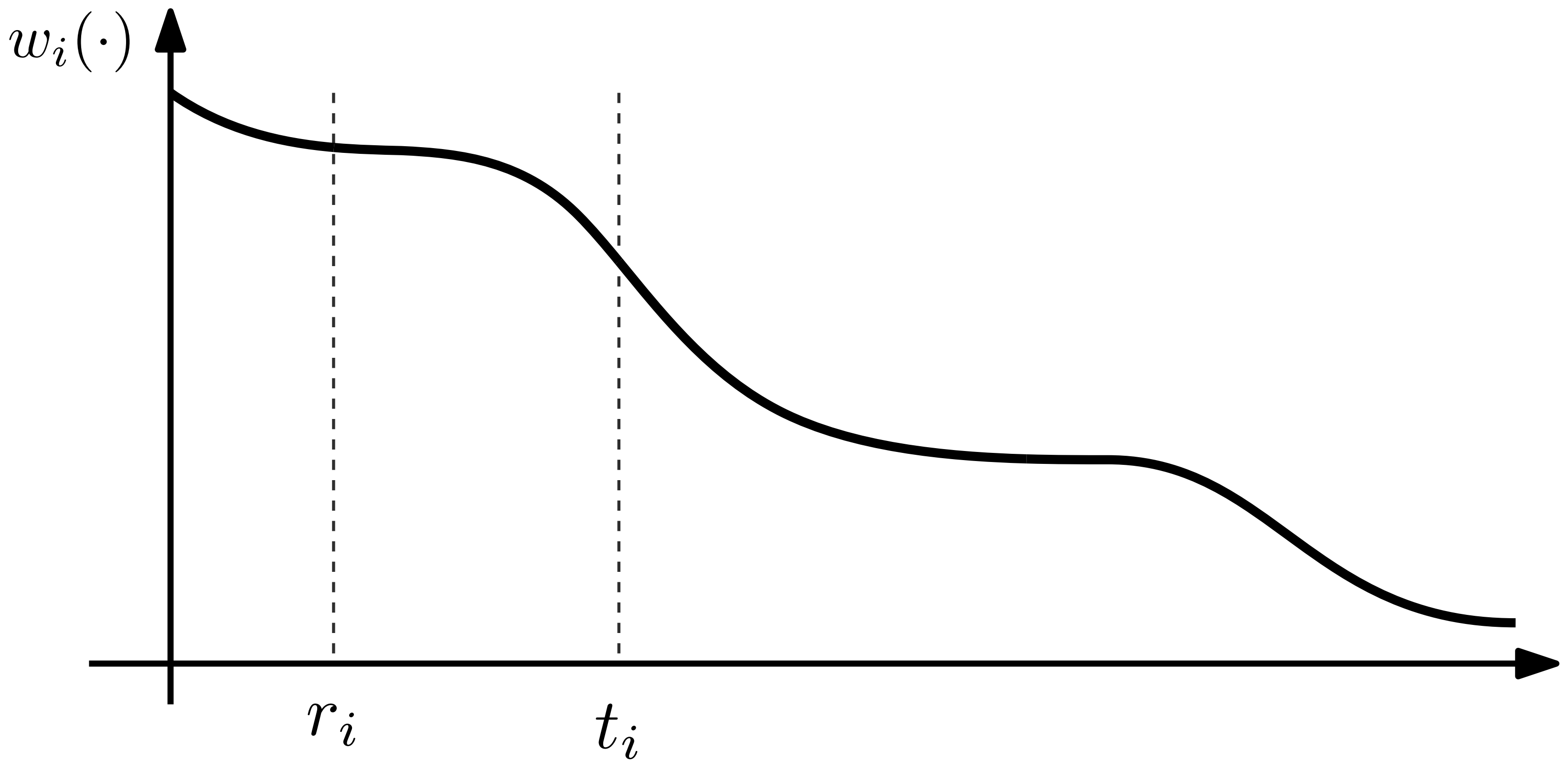
- When $r_i = t_i$: $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$
- When $r_i < t_i$:

$$u_i(t_i, g(r_{-i}, t_i)) = -(t_i - r_i)w_i(r_{-i}, r_i) - \int_0^{r_i} w_i(r_{-i}, z)dz$$

Proof (cont.):

- When $r_i = t_i$: $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$
- When $r_i < t_i$:

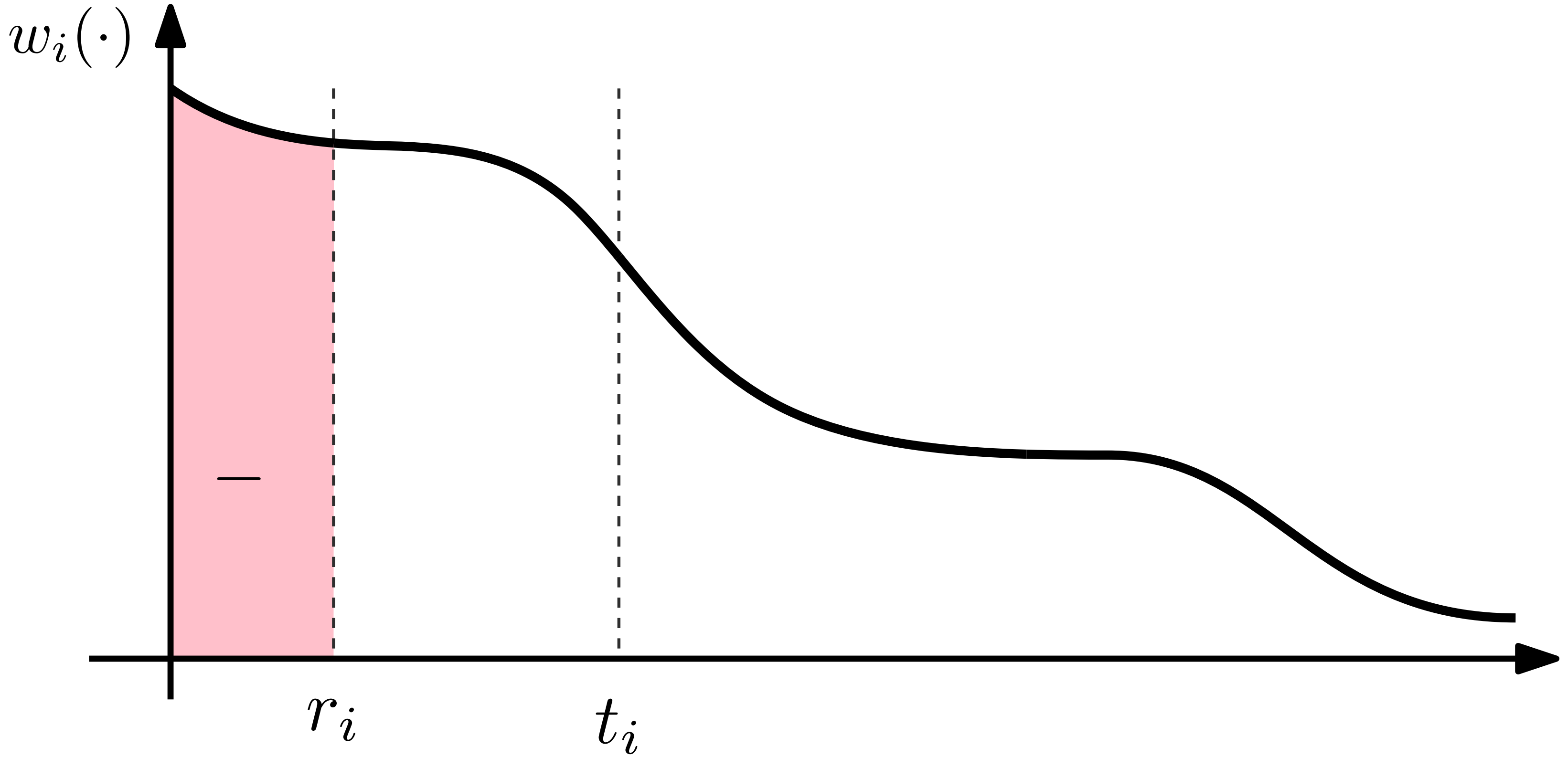
$$u_i(t_i, g(r_{-i}, t_i)) = -(t_i - r_i)w_i(r_{-i}, r_i) - \int_0^{r_i} w_i(r_{-i}, z)dz$$



Proof (cont.):

- When $r_i = t_i$: $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$
- When $r_i < t_i$:

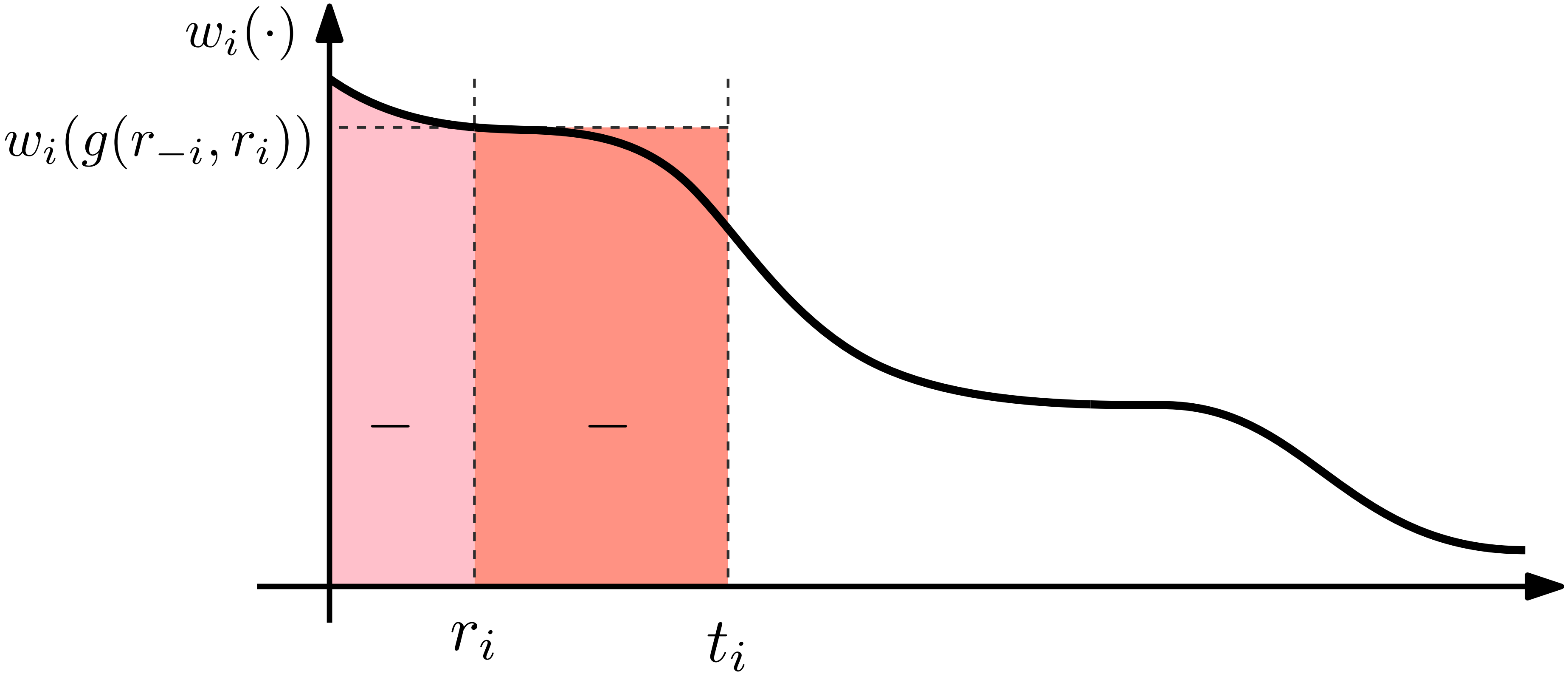
$$u_i(t_i, g(r_{-i}, t_i)) = -(t_i - r_i)w_i(r_{-i}, r_i) - \int_0^{r_i} w_i(r_{-i}, z)dz$$



Proof (cont.):

- When $r_i = t_i$: $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$
- When $r_i < t_i$:

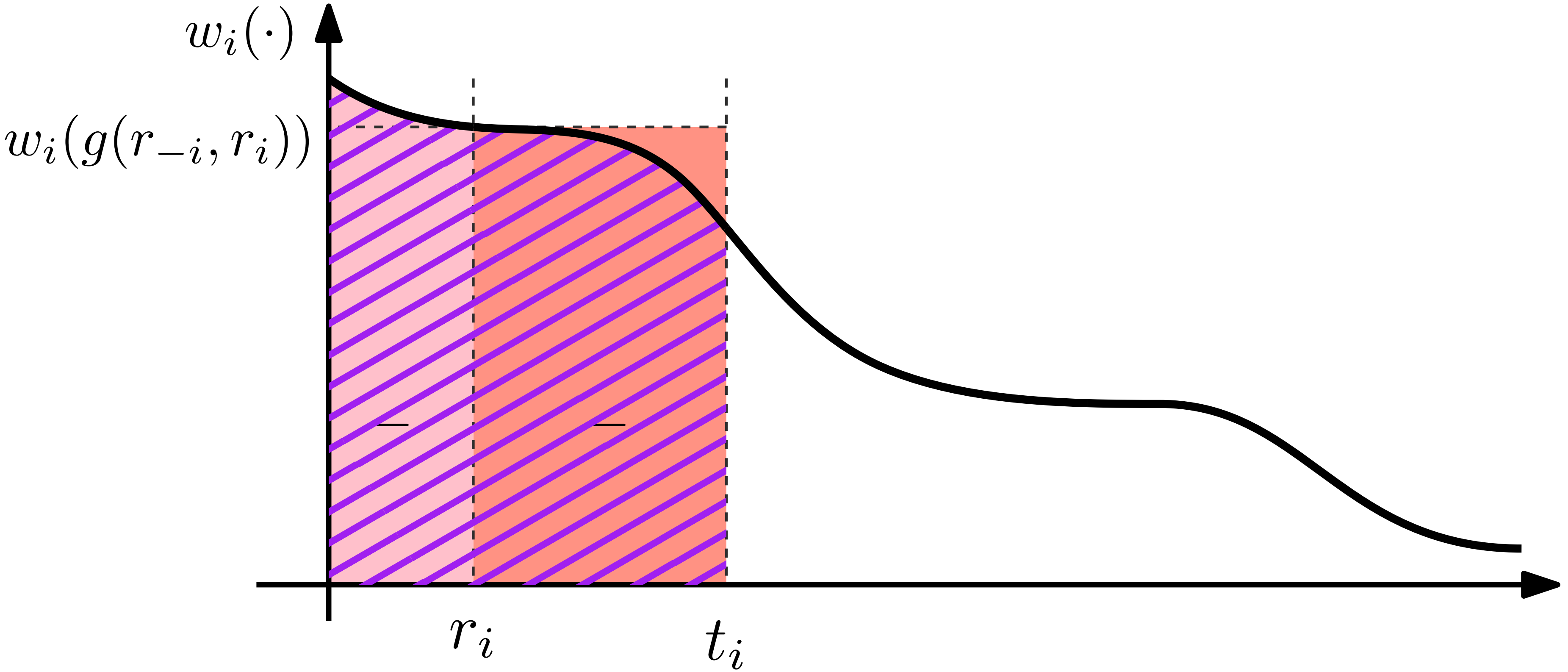
$$u_i(t_i, g(r_{-i}, t_i)) = \underline{-(t_i - r_i)w_i(r_{-i}, r_i)}\underline{-\int_0^{r_i} w_i(r_{-i}, z)dz}$$



Proof (cont.):

- When $r_i = t_i$: $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$
- When $r_i < t_i$:

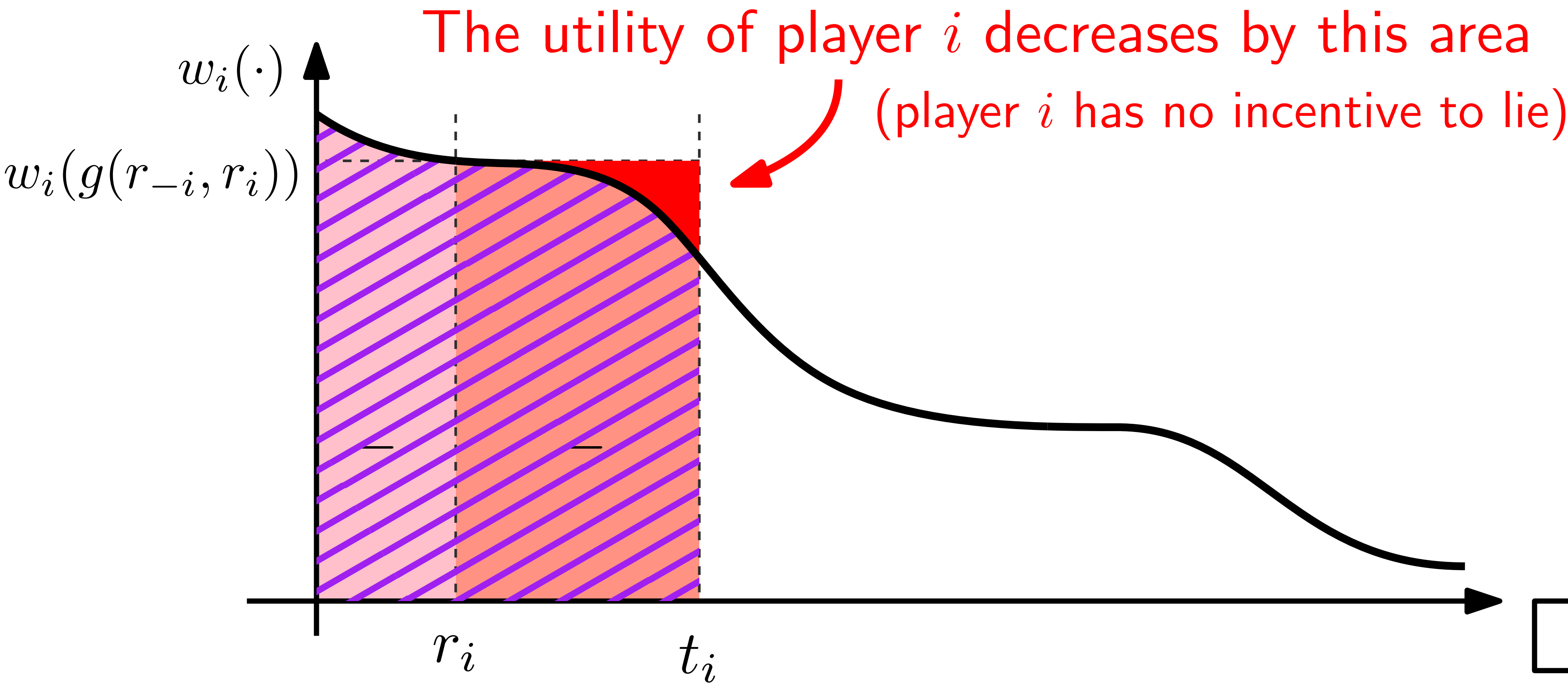
$$u_i(t_i, g(r_{-i}, t_i)) = -(t_i - r_i)w_i(r_{-i}, r_i) - \int_0^{r_i} w_i(r_{-i}, z)dz$$



Proof (cont.):

- When $r_i = t_i$: $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)dz$
- When $r_i < t_i$:

$$u_i(t_i, g(r_{-i}, t_i)) = -(t_i - r_i)w_i(r_{-i}, r_i) - \int_0^{r_i} w_i(r_{-i}, z)dz$$

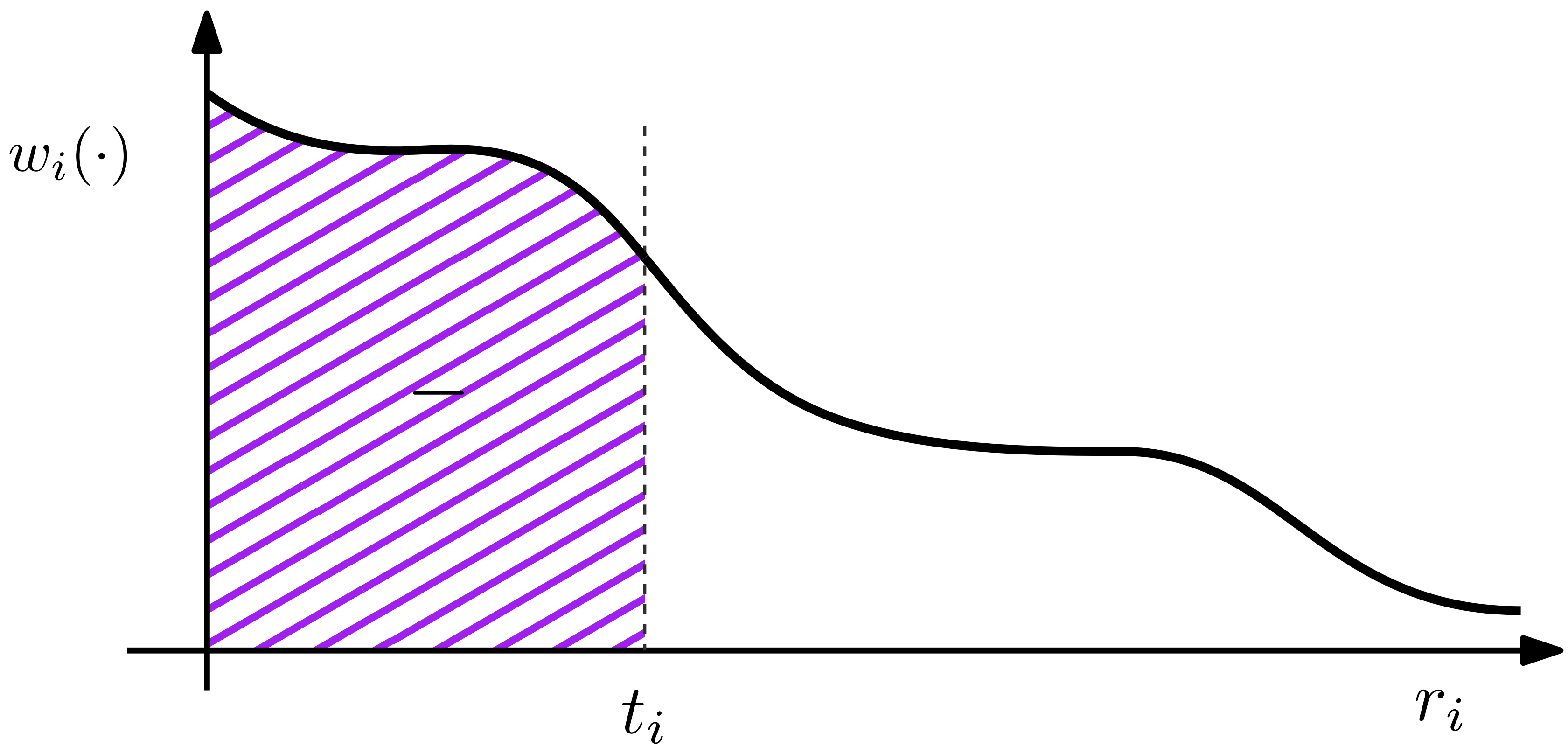


$\square$

Voluntary participation

With $h_i(r_{-i}) = 0$ the mechanism does not guarantee voluntary participation.

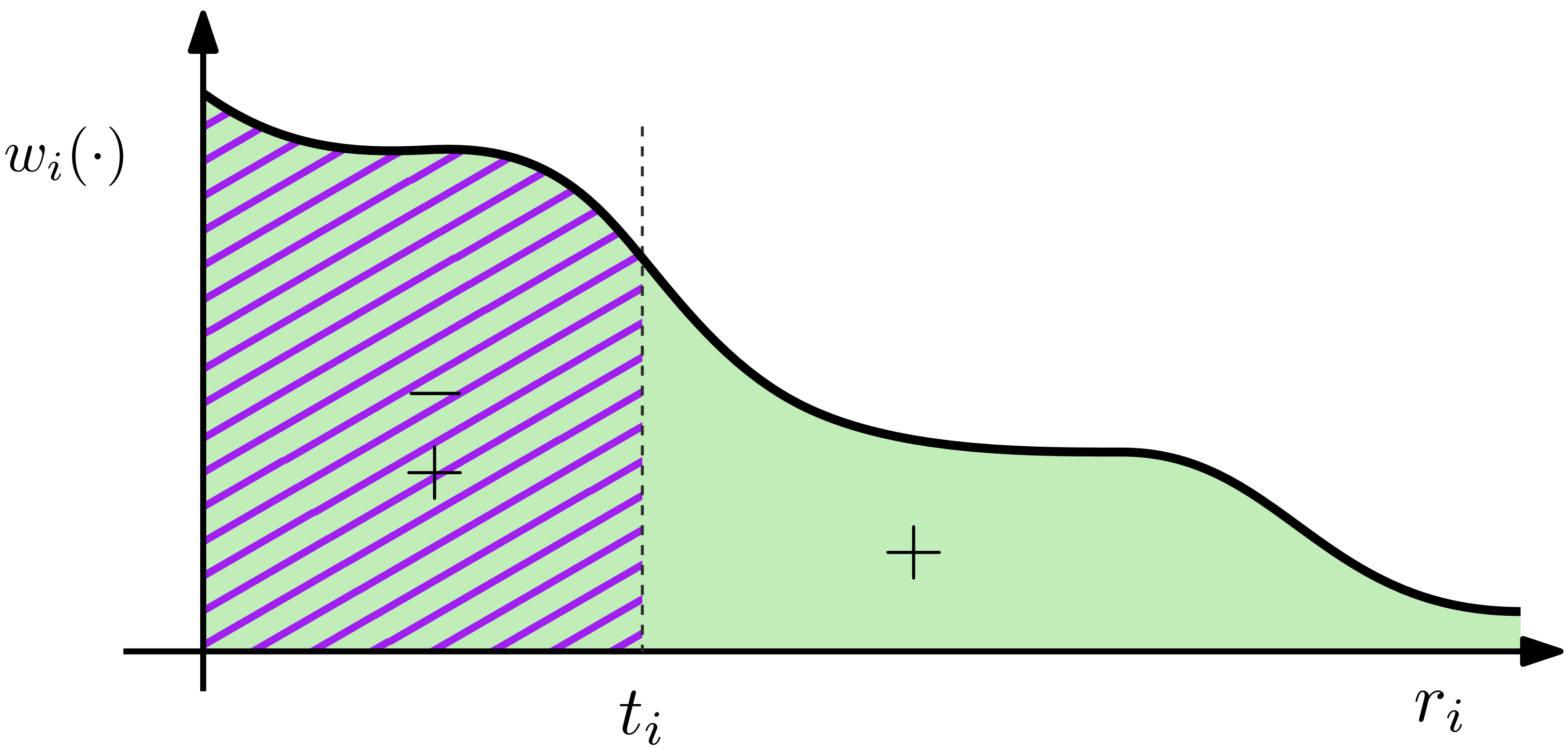
$$u_i(t_i, g(r_{-i}, t_i)) = \underbrace{h_i(r_{-i})}_{0} - \int_0^{t_i} w_i(r_{-i}, z)dz$$



Voluntary participation

Solution: Choose $h_i(r_{-i}) = \int_0^{+\infty} w_i(r_{-i}, z)dz$

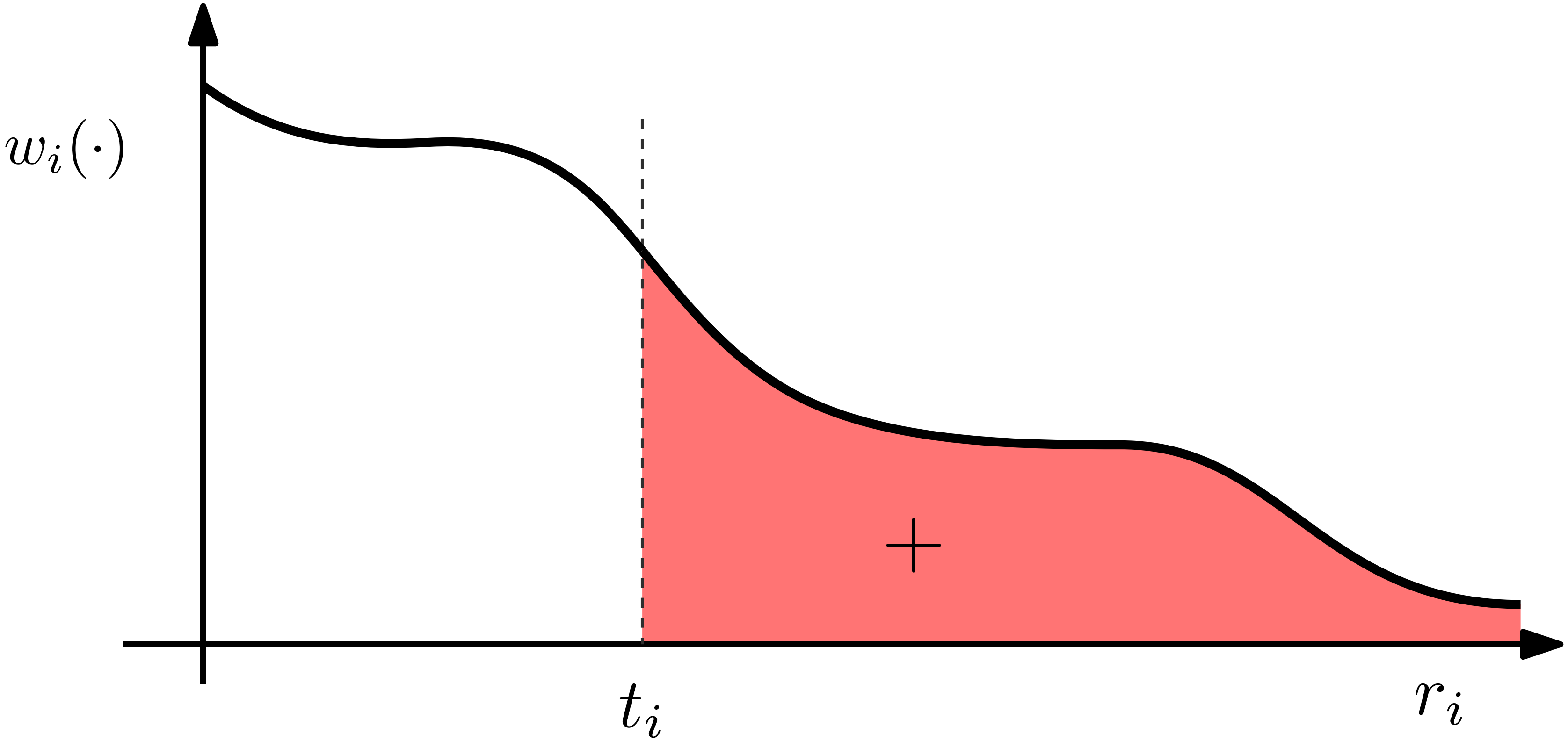
$$u_i(t_i, g(r_{-i}, t_i)) = h_i(r_{-i}) - \int_0^{t_i} w_i(r_{-i}, z)dz$$



Voluntary participation

Solution: Choose $h_i(r_{-i}) = \int_0^{+\infty} w_i(r_{-i}, z)dz$

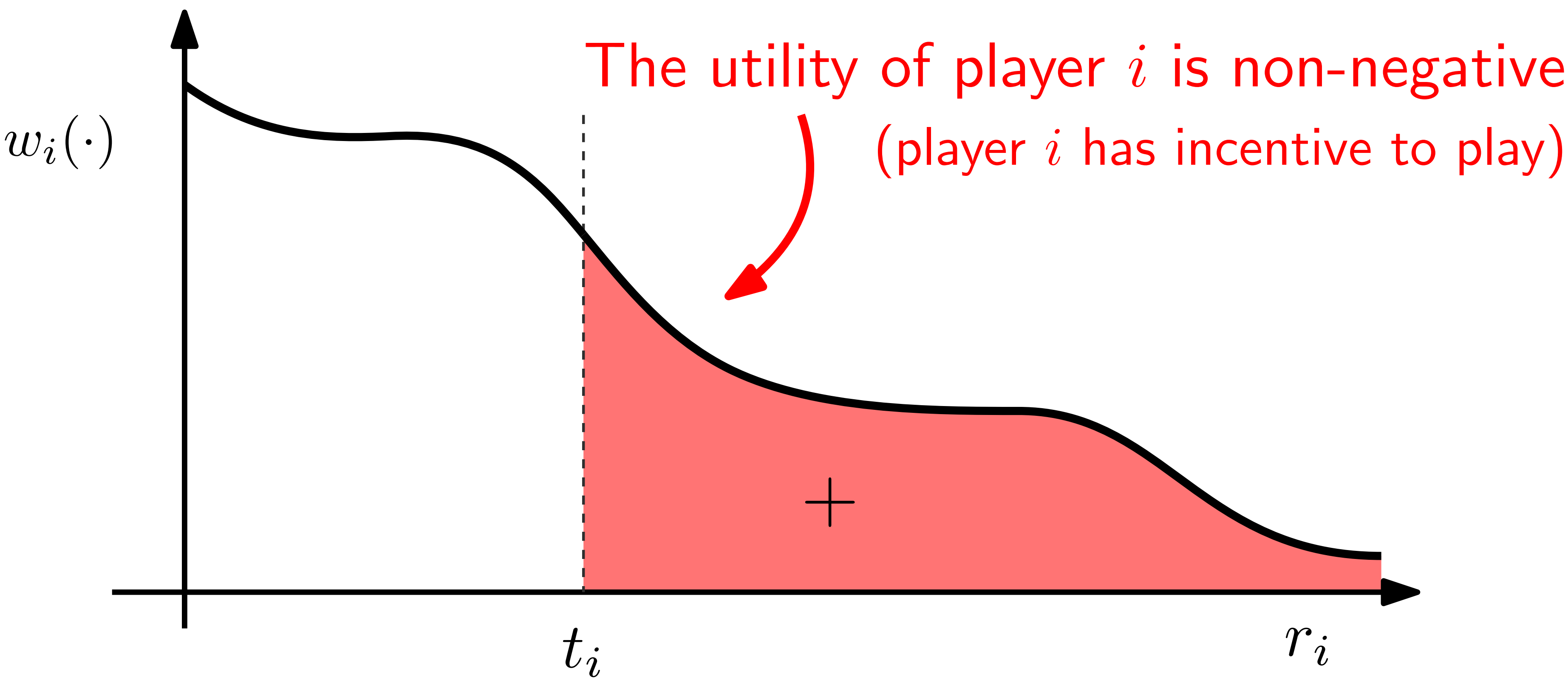
$$u_i(t_i, g(r_{-i}, t_i)) = \int_{r_i}^{+\infty} w_i(r_{-i}, z)dz$$



Voluntary participation

Solution: Choose $h_i(r_{-i}) = \int_0^{+\infty} w_i(r_{-i}, z)dz$

$$u_i(t_i, g(r_{-i}, t_i)) = \underline{\int_{r_i}^{+\infty} w_i(r_{-i}, z)dz}$$



Wrapping up

Truthful One-parameter mechanism that guarantees voluntary participation (for an OP problem):

$$M = \langle g, p \rangle$$

- g is any monotone algorithm (for the underlying OP problem)
- $p_i(r) = r_i w_i(r) + \int_{r_i}^{+\infty} w_i(r_{-i}, z) dz$

VCG vs One-parameter

VCG mechanisms: arbitrary valuation functions and types, but only utilitarian problems

One-parameter mechanisms: arbitrary social-choice function, but only one-parameter types and workloaded valuation functions

VCG vs One-parameter

VCG mechanisms: arbitrary valuation functions and types, but only utilitarian problems

One-parameter mechanisms: arbitrary social-choice function, but only one-parameter types and workloaded valuation functions

Note: A problem can be both utilitarian and One-parameter

VCG vs One-parameter

VCG mechanisms: arbitrary valuation functions and types, but only utilitarian problems

One-parameter mechanisms: arbitrary social-choice function, but only one-parameter types and workloaded valuation functions

Note: A problem can be both utilitarian and One-parameter

$$\Downarrow$$

The VCG and the OP mechanisms coincide