Patterned non-determinism in communication complexity

(Results and applications)

Dmytro Gavinsky

Institute of Mathematics, Praha
Czech Academy of Sciences

In 2022 the speaker changed the English spelling of his first name from the previous russian-odoured form *"Dmitry"* to the Ukrainian *"Dmytro"*.

Communication complexity

The setting of ***communication complexity*** is rather old: it was introduced by Abelson in 1977. To this day, it remains one of the most interesting computational models:

Communication complexity

The setting of ***communication complexity*** is rather old: it was introduced by Abelson in 1977. To this day, it remains one of the most interesting computational models:

- It is one of the strongest settings where we are able to prove "hardness" – that is, to establish *lower bounds* (often tight).

Communication complexity

The setting of ***communication complexity*** is rather old: it was introduced by Abelson in 1977. To this day, it remains one of the most interesting computational models:

- It is one of the strongest settings where we are able to prove "hardness" – that is, to establish *lower bounds* (often tight).
  On the other hand, it is one of the weakest settings where we can design arguably non-trivial algorithms – *communication protocols*.

Communication complexity

The setting of ***communication complexity*** is rather old: it was introduced by Abelson in 1977. To this day, it remains one of the most interesting computational models:

- It is one of the strongest settings where we are able to prove "hardness" – that is, to establish *lower bounds* (often tight).
  On the other hand, it is one of the weakest settings where we can design arguably non-trivial algorithms – *communication protocols*.
  Therefore, the communication complexity setting is one of those few that are *both "powerful" and "understandable"* enough to be interesting.

Communication complexity

The setting of ***communication complexity*** is rather old: it was introduced by Abelson in 1977. To this day, it remains one of the most interesting computational models:

- It is one of the strongest settings where we are able to prove "hardness" – that is, to establish *lower bounds* (often tight).
  On the other hand, it is one of the weakest settings where we can design arguably non-trivial algorithms – *communication protocols*.
  Therefore, the communication complexity setting is one of those few that are *both "powerful" and "understandable"* enough to be interesting.
- We can often compare the "*strength*" of two communication regimes via presenting a problem with an efficient solution in one, but not in the other. This can lead to non-trivial unconditional *structural separations* – that is to statements that certain tasks are *efficiently solvable* in one regime of communication but not in the other.

Communication complexity

The setting of ***communication complexity*** is rather old: it was introduced by Abelson in 1977. To this day, it remains one of the most interesting computational models:

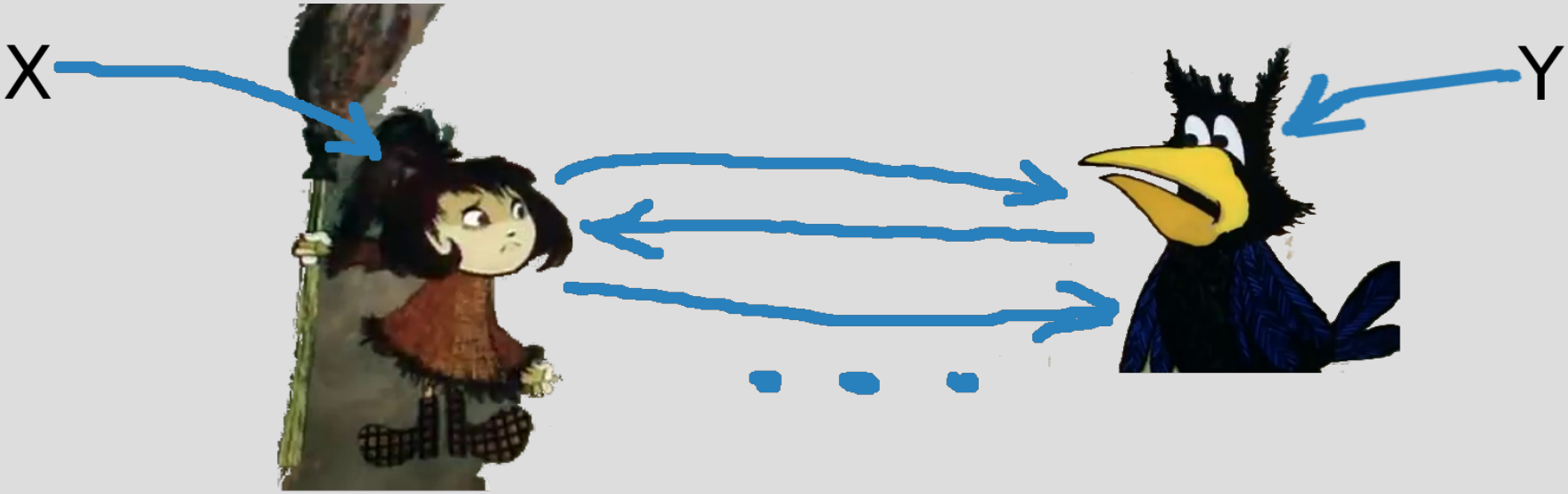
- It is one of the strongest settings where we are able to prove "hardness" – that is, to establish *lower bounds* (often tight).
  On the other hand, it is one of the weakest settings where we can design arguably non-trivial algorithms – *communication protocols*.
  Therefore, the communication complexity setting is one of those few that are *both "powerful" and "understandable"* enough to be interesting.
- We can often compare the "*strength*" of two communication regimes via presenting a problem with an efficient solution in one, but not in the other. This can lead to non-trivial unconditional *structural separations* – that is to statements that certain tasks are *efficiently solvable* in one regime of communication but not in the other.
- During *this talk* we will define and investigate a new model of ***non-deterministic communication***, which we will call ***patterned non-determinism*** (*PNP*).

Deterministic communication (Boolean case)

Deterministic communication (Boolean case)

- *Alice* receives X and *Bob* receives Y.

Deterministic communication (Boolean case)



- *Alice* receives X and *Bob* receives Y.
- They speak.

Deterministic communication (Boolean case)

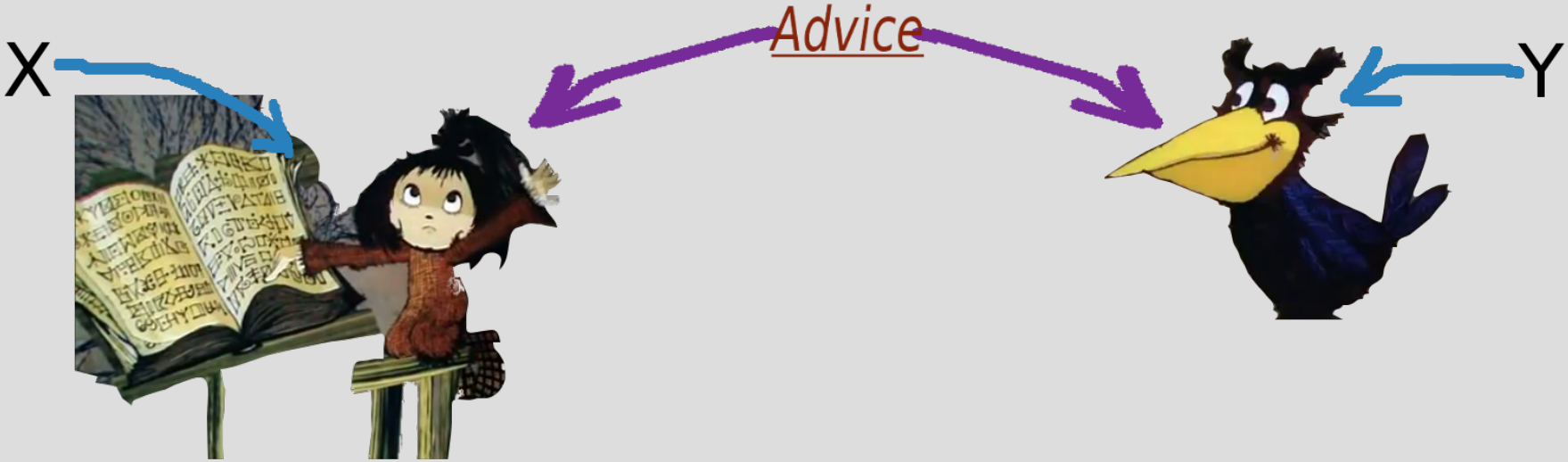
- *Alice* receives X and *Bob* receives Y.
- They speak.
- Bob *either accepts or rejects* the input (X, Y).

Non-deterministic communication

Non-deterministic communication

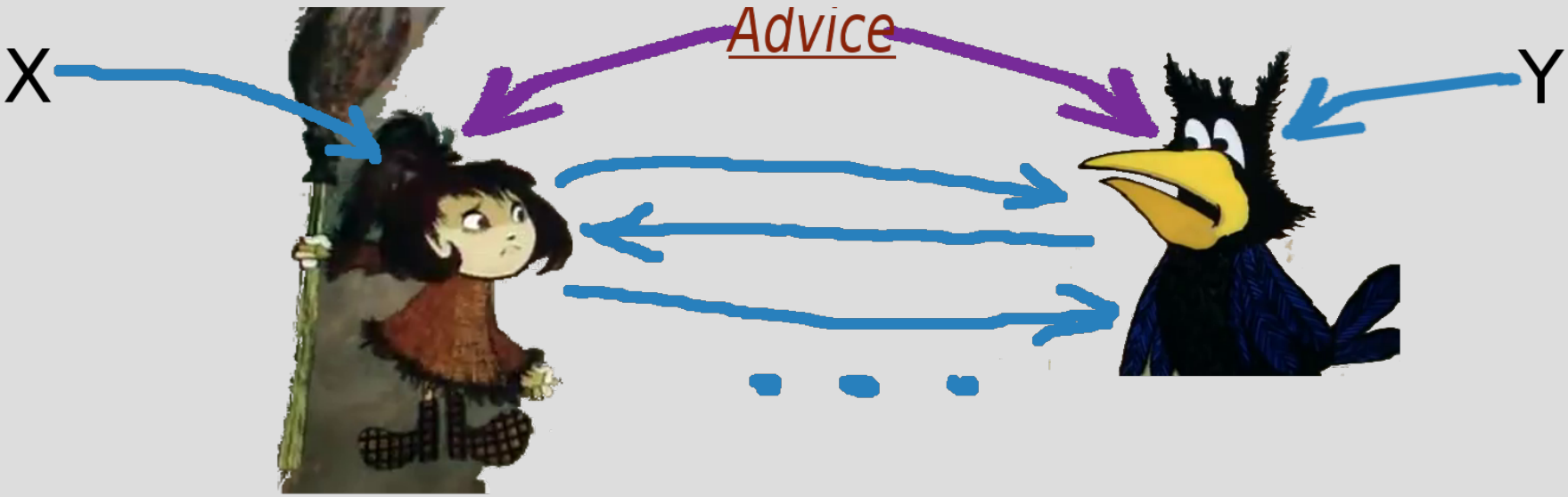
- *Alice* receives X and *Bob* receives Y.

Non-deterministic communication



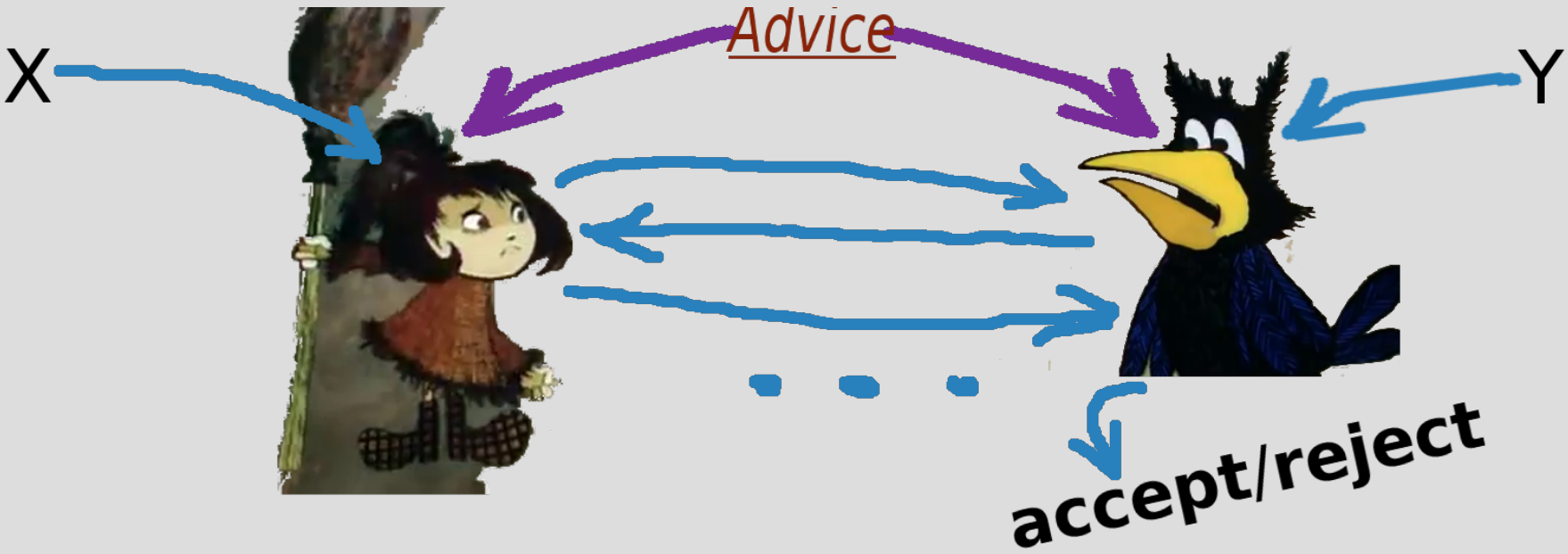
- *Alice* receives X and *Bob* receives Y.
- They both receive a *non-deterministic advice*.

Non-deterministic communication



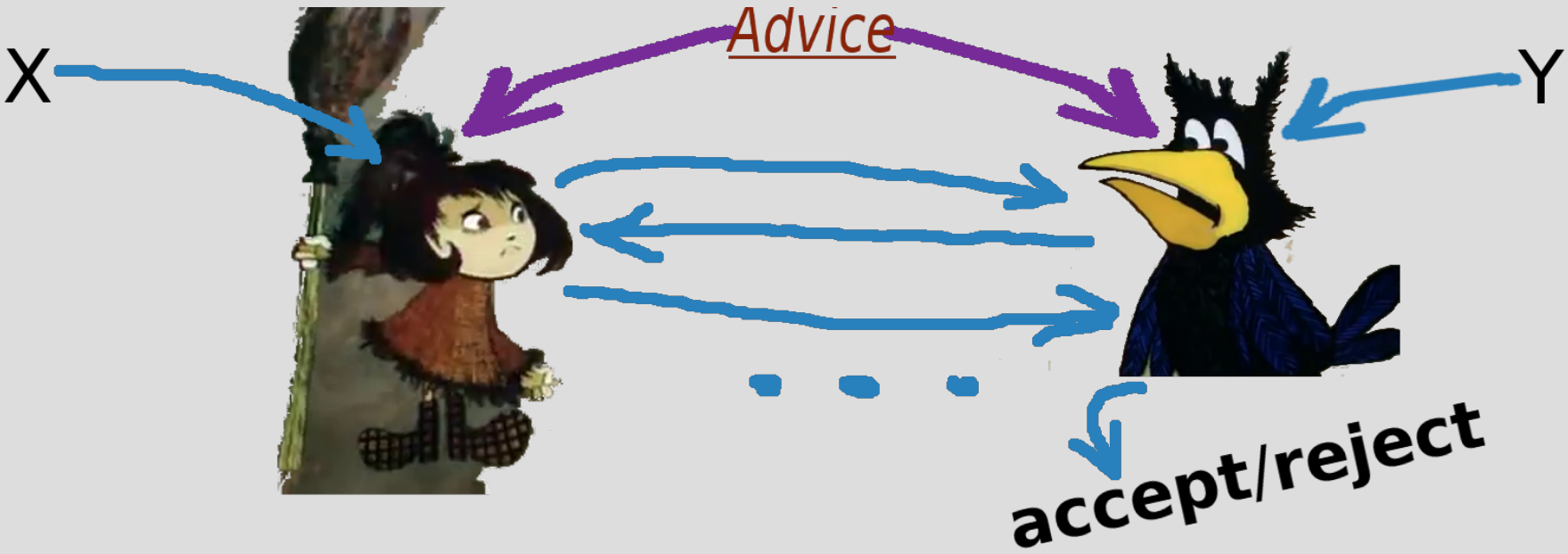
- *Alice* receives X and *Bob* receives Y.
- They both receive a *non-deterministic advice*.
- They speak.

Non-deterministic communication



- *Alice* receives X and *Bob* receives Y.
- They both receive a *non-deterministic advice*.
- They speak.
- Bob *either accepts or rejects* the input (X, Y) (under the given advice).

Non-deterministic communication



- *Alice* receives X and *Bob* receives Y.
- They both receive a *non-deterministic advice*.
- They speak.
- Bob *either accepts or rejects* the input (X, Y) (under the given advice).

The input pair is accepted by a non-deterministic protocol if *at least one advice value* leads to its acceptance.

Notions of efficiency

- A *deterministic protocol* over $(X, Y) \in \{0,1\}^n \times \{0,1\}^n$ is considered *efficient* if the players exchange *at most* poly-log(n) *bits*.

Notions of efficiency

- A *deterministic protocol* over $(X, Y) \in \{0,1\}^n \times \{0,1\}^n$ is considered *efficient* if the players exchange *at most* poly-log(n) *bits*.
- A *non-deterministic protocol* over $(X, Y) \in \{0,1\}^n \times \{0,1\}^n$ is considered *efficient* if *both the length of the advice and the number of exchanged bits* are *at most* poly-log(n).

Notions of efficiency

- A *deterministic protocol* over $(X, Y) \in \{0,1\}^n \times \{0,1\}^n$ is considered *efficient* if the players exchange *at most* poly-log(n) *bits*.
- A *non-deterministic protocol* over $(X, Y) \in \{0,1\}^n \times \{0,1\}^n$ is considered *efficient* if *both the length of the advice and the number of exchanged bits* are *at most* poly-log(n).

  Alternatively, $f : \{0,1\}^n \times \{0,1\}^n \to \{\top, \bot\}$ has an *efficient non-deterministic protocol* if it admits a decomposition of the form

  $$f(x, y) \equiv \bigvee_{i=1}^{m} f_i(x, y),$$

  where *m is at most quasi-polynomial in n* and *every f_i has an efficient deterministic protocol*: In this case a legitimate *advice* for input (x, y) would be any index i_0 such that $f_{i_0}(x, y) = \top$.

Notions of efficiency

- A *deterministic protocol* over $(X, Y) \in \{0,1\}^n \times \{0,1\}^n$ is considered *efficient* if the players exchange *at most* poly-log(n) *bits*.
- A *non-deterministic protocol* over $(X, Y) \in \{0,1\}^n \times \{0,1\}^n$ is considered *efficient* if *both the length of the advice and the number of exchanged bits* are *at most* poly-log(n).

  Alternatively, $f : \{0,1\}^n \times \{0,1\}^n \to \{\top, \bot\}$ has an *efficient non-deterministic protocol* if it admits a decomposition of the form

  $$f(x, y) \equiv \bigvee_{i=1}^{m} f_i(x, y),$$

  where *m is at most quasi-polynomial in n* and *every f_i has an efficient deterministic protocol*: In this case a legitimate *advice* for input (x, y) would be any index i_0 such that $f_{i_0}(x, y) = \top$.
- Clearly, *non-deterministic protocols are at least as strong as deterministic ones*; on the other hand, there are *functions with very efficient non-deterministic protocols but with deterministic complexity in $\Omega(n)$*.

Restricted non-determinism, old and new models

N.b. We will use the same notation (P, NP, etc.) for both the communication models and the corresponding classes of problems with efficient protocols.

Restricted non-determinism, old and new models

N.b. We will use the same notation (P, NP, etc.) for both the communication models and the corresponding classes of problems with efficient protocols.

- Denote, respectively, by ***P*** and ***NP*** the models of deterministic and non-deterministic communication.

Restricted non-determinism, old and new models

N.b. We will use the same notation (P, NP, etc.) for both the communication models and the corresponding classes of problems with efficient protocols.

- Denote, respectively, by P and NP the models of deterministic and non-deterministic communication.
- Two classes of interest to us have been defined "*between P and NP*":

Restricted non-determinism, old and new models

N.b. We will use the same notation (P, NP, etc.) for both the communication models and the corresponding classes of problems with efficient protocols.

- Denote, respectively, by ***P*** and ***NP*** the models of deterministic and non-deterministic communication.
- Two classes of interest to us have been defined "*between P and NP*":
  - ***UP*** is the sub-class of NP, where every input has *at most one* advice value that leads to its acceptance; *alternatively*, $f \in UP$ if it has an NP-decomposition $f(x,y) = \vee_{i=1}^{m} f_i(x,y)$ is such that $\forall\, x, y$: $\left|\left\{i \middle| f_i(x,y) = \top\right\}\right| \leq 1$.

Restricted non-determinism, old and new models

N.b. We will use the same notation (P, NP, etc.) for both the communication models and the corresponding classes of problems with efficient protocols.

- Denote, respectively, by ***P*** and ***NP*** the models of deterministic and non-deterministic communication.
- Two classes of interest to us have been defined "*between P and NP*":
  - ***UP*** is the sub-class of NP, where every input has *at most one* advice value that leads to its acceptance; *alternatively*, $f \in UP$ if it has an NP-decomposition $f(x,y) = \vee_{i=1}^{m} f_i(x,y)$ is such that $\forall\, x, y$: $\left|\left\{i \middle| f_i(x,y) = \top\right\}\right| \leq 1$.
  - ***FewP*** is the sub-class of NP, where every input has *at most* $\text{poly-log}(n)$ advice values leading to its acceptance; *alternatively*, $f \in FewP$ if $\forall\, x, y$: $\left|\left\{i \middle| f_i(x,y) = \top\right\}\right| \leq \text{poly-log}(n)$ in an NP-decomposition of $f(x,y)$.

Restricted non-determinism, old and new models

N.b. We will use the same notation (P, NP, etc.) for both the communication models and the corresponding classes of problems with efficient protocols.

- Denote, respectively, by P and NP the models of deterministic and non-deterministic communication.
- Two classes of interest to us have been defined "*between P and NP*":
  - UP is the sub-class of NP, where every input has *at most one* advice value that leads to its acceptance; *alternatively*, $f \in UP$ if it has an NP-decomposition $f(x,y) = \vee_{i=1}^{m} f_i(x,y)$ is such that $\forall\, x, y$: $\left|\left\{i \middle| f_i(x,y) = \top\right\}\right| \leq 1$.
  - $FewP$ is the sub-class of NP, where every input has *at most* poly-log(n) advice values leading to its acceptance; *alternatively*, $f \in FewP$ if $\forall\, x, y$: $\left|\left\{i \middle| f_i(x,y) = \top\right\}\right| \leq$ poly-log(n) in an NP-decomposition of $f(x,y)$.
- We define the model of **patterned non-determinism**, denoted PNP:

Restricted non-determinism, old and new models

N.b. We will use the same notation (P, NP, etc.) for both the communication models and the corresponding classes of problems with efficient protocols.

- Denote, respectively, by P and NP the models of deterministic and non-deterministic communication.
- Two classes of interest to us have been defined "*between P and NP*":
  - UP is the sub-class of NP, where every input has *at most one* advice value that leads to its acceptance; *alternatively*, $f \in UP$ if it has an NP-decomposition $f(x,y) = \vee_{i=1}^{m} f_i(x,y)$ is such that $\forall\, x, y$: $\left|\left\{i \middle| f_i(x,y) = \top\right\}\right| \leq 1$.
  - $FewP$ is the sub-class of NP, where every input has *at most* poly-log(n) advice values leading to its acceptance; *alternatively*, $f \in FewP$ if $\forall\, x, y$: $\left|\left\{i \middle| f_i(x,y) = \top\right\}\right| \leq$ poly-log(n) in an NP-decomposition of $f(x,y)$.
- We define the model of **patterned non-determinism**, denoted PNP:
  - For $f : \{0,1\}^n \times \{0,1\}^n \to \{\top, \bot\}$ with an NP-decomposition

    $$f(x,y) = \vee_{i=1}^{m} f_i(x,y),$$

    let the corresponding family of **accepting patterns** be defined as

    $$\Gamma_f \stackrel{\text{def}}{=} \left\{\left\{i \middle| f_i(x,y) = \top\right\} \middle| (x,y) \in \{0,1\}^n \times \{0,1\}^n\right\}.$$

Restricted non-determinism, old and new models

N.b. We will use the same notation (P, NP, etc.) for both the communication models and the corresponding classes of problems with efficient protocols.

- Denote, respectively, by P and NP the models of deterministic and non-deterministic communication.
- Two classes of interest to us have been defined "*between P and NP*":
  - UP is the sub-class of NP, where every input has *at most one* advice value that leads to its acceptance; *alternatively*, $f \in UP$ if it has an NP-decomposition $f(x,y) = \vee_{i=1}^{m} f_i(x,y)$ is such that $\forall\, x, y$: $\left|\left\{i \middle| f_i(x,y) = \top\right\}\right| \leq 1$.
  - $FewP$ is the sub-class of NP, where every input has *at most* poly-log(n) advice values leading to its acceptance; *alternatively*, $f \in FewP$ if $\forall\, x, y$: $\left|\left\{i \middle| f_i(x,y) = \top\right\}\right| \leq$ poly-log(n) in an NP-decomposition of $f(x,y)$.
- We define the model of **patterned non-determinism**, denoted PNP:
  - For $f : \{0,1\}^n \times \{0,1\}^n \to \{\top, \bot\}$ with an NP-decomposition
    $$f(x,y) = \vee_{i=1}^{m} f_i(x,y),$$
    let the corresponding family of **accepting patterns** be defined as
    $$\Gamma_f \stackrel{\text{def}}{=} \left\{\left\{i \middle| f_i(x,y) = \top\right\} \middle| (x,y) \in \{0,1\}^n \times \{0,1\}^n\right\}.$$
  - Then $f \in PNP$ if $|\Gamma_f|$ is *at most quasi-polynomial in* n.

Some relations among the defined models

- As mentioned earlier, $P \subset NP$: the proper containment is witnessed, in particular, by the *equality function*.

Some relations among the defined models

- As mentioned earlier, $P \subset NP$: the proper containment is witnessed, in particular, by the *equality function*.
- From the definitions,

$$P \subseteq UP \subseteq FewP \subseteq PNP \subseteq NP.$$

Some relations among the defined models

- As mentioned earlier, $P \subset NP$: the proper containment is witnessed, in particular, by the *equality function*.
- From the definitions,

$$P \subseteq UP \subseteq FewP \subseteq PNP \subseteq NP.$$

- For *total functions* – that is, any $f : \{0,1\}^n \times \{0,1\}^n \to \{\top, \bot\}$:

Some relations among the defined models

- As mentioned earlier, $P \subset NP$: the proper containment is witnessed, in particular, by the *equality function*.
- From the definitions,

$$P \subseteq UP \subseteq FewP \subseteq PNP \subseteq NP.$$

- For *total functions* – that is, any $f : \{0,1\}^n \times \{0,1\}^n \to \{\top, \bot\}$:
  - Yannakakis [*Yan91*] proved that $UP = P$.

Some relations among the defined models

- As mentioned earlier, $P \subset NP$: the proper containment is witnessed, in particular, by the *equality function*.
- From the definitions,

$$P \subseteq UP \subseteq FewP \subseteq PNP \subseteq NP.$$

- For *total functions* – that is, any $f : \{0,1\}^n \times \{0,1\}^n \to \{\top, \bot\}$:
  - Yannakakis [*Yan91*] proved that $UP = P$.
  - Karchmer, Newman, Saks and Wigderson [*KNSW94*] strengthened the above to $FewP = P$.

Some relations among the defined models

- As mentioned earlier, $P \subset NP$: the proper containment is witnessed, in particular, by the *equality function*.
- From the definitions,

$$P \subseteq UP \subseteq FewP \subseteq PNP \subseteq NP.$$

- For *total functions* – that is, any $f : \{0,1\}^n \times \{0,1\}^n \to \{\top, \bot\}$:
  - Yannakakis [*Yan91*] proved that $UP = P$.
  - Karchmer, Newman, Saks and Wigderson [*KNSW94*] strengthened the above to $FewP = P$.
  - We show (the argument is omitted from this presentation) that $PNP = P$.

Some relations among the defined models

- As mentioned earlier, $P \subset NP$: the proper containment is witnessed, in particular, by the *equality function*.
- From the definitions,

$$P \subseteq UP \subseteq FewP \subseteq PNP \subseteq NP.$$

- For *total functions* – that is, any $f : \{0,1\}^n \times \{0,1\}^n \to \{\top, \bot\}$:
  - Yannakakis [*Yan91*] proved that $UP = P$.
  - Karchmer, Newman, Saks and Wigderson [*KNSW94*] strengthened the above to $FewP = P$.
  - We show (the argument is omitted from this presentation) that $PNP = P$.
  - That is,

$$P = UP = FewP = PNP \subset NP.$$

Some relations among the defined models

- As mentioned earlier, $P \subset NP$: the proper containment is witnessed, in particular, by the *equality function*.
- From the definitions,

$$P \subseteq UP \subseteq FewP \subseteq PNP \subseteq NP.$$

- For *total functions* – that is, any $f : \{0,1\}^n \times \{0,1\}^n \to \{\top, \bot\}$:
  - Yannakakis [*Yan91*] proved that $UP = P$.
  - Karchmer, Newman, Saks and Wigderson [*KNSW94*] strengthened the above to $FewP = P$.
  - We show (the argument is omitted from this presentation) that $PNP = P$.
  - That is,

$$P = UP = FewP = PNP \subset NP.$$

  - *"Totality" is crucial* for these model equivalences: for *partial functions* even $UP \neq P$.

The pattern search problem

- When "$UP = P$" or "$FewP = P$" or "$PNP = P$" – that is, that the corresponding case of *restricted non-determinism* is not stronger than mere determinism – one can ask whether **an advice value** that witnesses $[f(x, y) = \top]$ *can be found efficiently by a deterministic protocol* for every $(x, y) \in f^{-1}(\top)$ for each f in UP, $FewP$ or PNP, respectively.

The pattern search problem

- When "$UP = P$" or "$FewP = P$" or "$PNP = P$" – that is, that the corresponding case of *restricted non-determinism* is not stronger than mere determinism – one can ask whether **an advice value** that witnesses $[f(x, y) = \top]$ *can be found efficiently by a deterministic protocol* for every $(x, y) \in f^{-1}(\top)$ for each f in UP, $FewP$ or PNP, respectively.

  The answer is *affirmative*, as in each of these cases simple *binary search* efficiently produces a legitimate witness.

The pattern search problem

- When "$UP = P$" or "$FewP = P$" or "$PNP = P$" – that is, that the corresponding case of *restricted non-determinism* is not stronger than mere determinism – one can ask whether **an advice value** that witnesses $[f(x, y) = \top]$ *can be found efficiently by a deterministic protocol* for every $(x, y) \in f^{-1}(\top)$ for each f in UP, $FewP$ or PNP, respectively.

  The answer is *affirmative*, as in each of these cases simple *binary search* efficiently produces a legitimate witness.

- Then one can ask whether the set of **all convincing witnesses** *can be found efficiently*: the answer is, trivially, affirmative for both UP and $FewP$, while *in the case of* PNP it may be somewhat less straightforward.

The pattern search problem

- When "$UP = P$" or "$FewP = P$" or "$PNP = P$" – that is, that the corresponding case of *restricted non-determinism* is not stronger than mere determinism – one can ask whether **an advice value** that witnesses $[f(x, y) = \top]$ *can be found efficiently by a deterministic protocol* for every $(x, y) \in f^{-1}(\top)$ for each f in UP, $FewP$ or PNP, respectively.

  The answer is *affirmative*, as in each of these cases simple *binary search* efficiently produces a legitimate witness.

- Then one can ask whether the set of **all convincing witnesses** *can be found efficiently*: the answer is, trivially, affirmative for both UP and $FewP$, while *in the case of PNP* it may be somewhat less straightforward.

  We show (the argument is omitted from this presentation) that the answer is *affirmative* – that is, the precise **accepting pattern** of every $(x, y) \in f^{-1}(\top)$ *can be found efficiently by a deterministic protocol* for each $f \in PNP$.

The pattern search problem

- When "$UP = P$" or "$FewP = P$" or "$PNP = P$" – that is, that the corresponding case of *restricted non-determinism* is not stronger than mere determinism – one can ask whether **an advice value** that witnesses $[f(x, y) = \top]$ *can be found efficiently by a deterministic protocol* for every $(x, y) \in f^{-1}(\top)$ for each f in UP, $FewP$ or PNP, respectively.

  The answer is *affirmative*, as in each of these cases simple *binary search* efficiently produces a legitimate witness.
- Then one can ask whether the set of **all convincing witnesses** *can be found efficiently*: the answer is, trivially, affirmative for both UP and $FewP$, while *in the case of PNP* it may be somewhat less straightforward.

  We show (the argument is omitted from this presentation) that the answer is *affirmative* – that is, the precise **accepting pattern** of every $(x, y) \in f^{-1}(\top)$ *can be found efficiently by a deterministic protocol* for each $f \in PNP$.
- We shall see next how the above statement leads to certain (possibly, surprising) model equivalence in *multi-party communication complexity*.

Three-party communication with listening Charlie

Consider the model

$$(\textbf{Alice} \leftrightarrow \textbf{Bob}) \rightarrow \textbf{Charlie}.$$

Three-party communication with listening Charlie

Consider the model

$$(\textbf{Alice} \leftrightarrow \textbf{Bob}) \rightarrow \textbf{Charlie}.$$

That is,

- *Alice* receives X, *Bob* receives Y and *Charlie* receives Z;

Three-party communication with listening Charlie

Consider the model

$$(\textbf{Alice} \leftrightarrow \textbf{Bob}) \rightarrow \textbf{Charlie}.$$

That is,

- *Alice* receives X, *Bob* receives Y and *Charlie* receives Z;
- Alice speaks with Bob in order to produce a single message for Charlie;

Three-party communication with listening Charlie

Consider the model

$$(\textbf{Alice} \leftrightarrow \textbf{Bob}) \rightarrow \textbf{Charlie}.$$

That is,

- *Alice* receives X, *Bob* receives Y and *Charlie* receives Z;
- Alice speaks with Bob in order to produce a single message for Charlie;
- Charlie receives the message and outputs the answer with respect to the input (X, Y, Z).

Three-party communication with listening Charlie

Consider the model

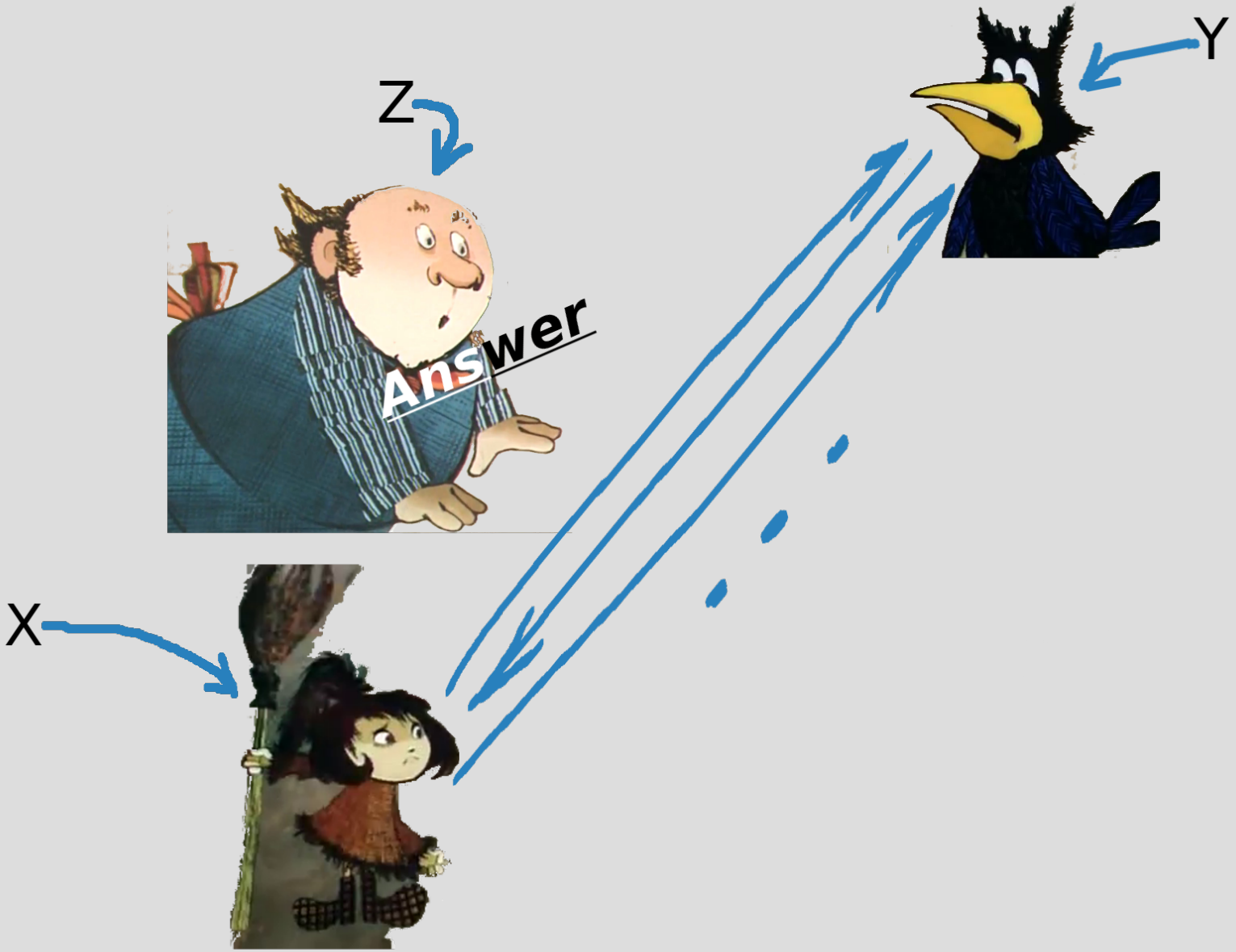
$$(\textbf{Alice} \leftrightarrow \textbf{Bob}) \rightarrow \textbf{Charlie}.$$

That is,

- *Alice* receives X, *Bob* receives Y and *Charlie* receives Z;
- Alice speaks with Bob in order to produce a single message for Charlie;
- Charlie receives the message and outputs the answer with respect to the input (X, Y, Z).

  Alternatively, Alice interacts with Bob; Charlie sees the full transcript of their conversation and produces the answer.

Three-party communication with listening Charlie



Alice speaks with Bob; Charlie hears them and produces the answer.

Model equivalence

- For brevity, let us denote the model of *three-party communication with listening Charlie* by $[(X \leftrightarrow Y) \rightarrow Z]$.

Model equivalence

- For brevity, let us denote the model of *three-party communication with listening Charlie* by $[(X \leftrightarrow Y) \rightarrow Z]$.
- Let $[(X, Z) \leftrightarrow Y]$ denote the model where *Player I* receives (X, Z), *Player II* receives Y and they interact in order to compute $f(X, Y, Z)$.

Model equivalence

- For brevity, let us denote the model of *three-party communication with listening Charlie* by $[(X \leftrightarrow Y) \rightarrow Z]$.
- Let $[(X, Z) \leftrightarrow Y]$ denote the model where *Player I* receives (X, Z), *Player II* receives Y and they interact in order to compute $f(X, Y, Z)$. Obviously, $[(X, Z) \leftrightarrow Y]$ *is at least as strong as* $[(X \leftrightarrow Y) \rightarrow Z]$.

Model equivalence

- For brevity, let us denote the model of *three-party communication with listening Charlie* by $[(X \leftrightarrow Y) \rightarrow Z]$.
- Let $[(X, Z) \leftrightarrow Y]$ denote the model where *Player I* receives (X, Z), *Player II* receives Y and they interact in order to compute $f(X, Y, Z)$. Obviously, $[(X, Z) \leftrightarrow Y]$ *is at least as strong as* $[(X \leftrightarrow Y) \rightarrow Z]$.
- Let $[(X, Y) \rightarrow Z]$ denote the model where *Player I* receives (X, Y), *Player II* receives Z and Player I sends a *single message* to Player II to let him compute the value of $f(X, Y, Z)$.

Model equivalence

- For brevity, let us denote the model of *three-party communication with listening Charlie* by $[(X \leftrightarrow Y) \rightarrow Z]$.
- Let $[(X, Z) \leftrightarrow Y]$ denote the model where *Player I* receives (X, Z), *Player II* receives Y and they interact in order to compute $f(X, Y, Z)$. Obviously, $[(X, Z) \leftrightarrow Y]$ *is at least as strong as* $[(X \leftrightarrow Y) \rightarrow Z]$.
- Let $[(X, Y) \rightarrow Z]$ denote the model where *Player I* receives (X, Y), *Player II* receives Z and Player I sends a *single message* to Player II to let him compute the value of $f(X, Y, Z)$.
  Obviously, $[(X, Y) \rightarrow Z]$ *is at least as strong as* $[(X \leftrightarrow Y) \rightarrow Z]$.

Model equivalence

- For brevity, let us denote the model of *three-party communication with listening Charlie* by $[(X \leftrightarrow Y) \rightarrow Z]$.
- Let $[(X, Z) \leftrightarrow Y]$ denote the model where *Player I* receives (X, Z), *Player II* receives Y and they interact in order to compute $f(X, Y, Z)$.
  Obviously, $[(X, Z) \leftrightarrow Y]$ *is at least as strong as* $[(X \leftrightarrow Y) \rightarrow Z]$.
- Let $[(X, Y) \rightarrow Z]$ denote the model where *Player I* receives (X, Y), *Player II* receives Z and Player I sends a *single message* to Player II to let him compute the value of $f(X, Y, Z)$.
  Obviously, $[(X, Y) \rightarrow Z]$ *is at least as strong as* $[(X \leftrightarrow Y) \rightarrow Z]$.
- The bipartite models $[(X, Z) \leftrightarrow Y]$ and $[(X, Y) \rightarrow Z]$ are natural **two-party reductions** of the three-party model $[(X \leftrightarrow Y) \rightarrow Z]$.

Model equivalence

- For brevity, let us denote the model of *three-party communication with listening Charlie* by $[(X \leftrightarrow Y) \rightarrow Z]$.
- Let $[(X, Z) \leftrightarrow Y]$ denote the model where *Player I* receives (X, Z), *Player II* receives Y and they interact in order to compute $f(X, Y, Z)$.
  Obviously, $[(X, Z) \leftrightarrow Y]$ *is at least as strong as* $[(X \leftrightarrow Y) \rightarrow Z]$.
- Let $[(X, Y) \rightarrow Z]$ denote the model where *Player I* receives (X, Y), *Player II* receives Z and Player I sends a *single message* to Player II to let him compute the value of $f(X, Y, Z)$.
  Obviously, $[(X, Y) \rightarrow Z]$ *is at least as strong as* $[(X \leftrightarrow Y) \rightarrow Z]$.
- The bipartite models $[(X, Z) \leftrightarrow Y]$ and $[(X, Y) \rightarrow Z]$ are natural **two-party reductions** of the three-party model $[(X \leftrightarrow Y) \rightarrow Z]$.

We show that *the converse is also true* – that is, *for* ***total functions*** *the three-party model* $[(X \leftrightarrow Y) \rightarrow Z]$ *is as strong as the weaker* (with respect to the same function) *of its two-party reductions* $[(X, Z) \leftrightarrow Y]$ *and* $[(X, Y) \rightarrow Z]$.

Model equivalence

- For brevity, let us denote the model of *three-party communication with listening Charlie* by $[(X \leftrightarrow Y) \rightarrow Z]$.
- Let $[(X, Z) \leftrightarrow Y]$ denote the model where *Player I* receives (X, Z), *Player II* receives Y and they interact in order to compute $f(X, Y, Z)$.
  Obviously, $[(X, Z) \leftrightarrow Y]$ *is at least as strong as* $[(X \leftrightarrow Y) \rightarrow Z]$.
- Let $[(X, Y) \rightarrow Z]$ denote the model where *Player I* receives (X, Y), *Player II* receives Z and Player I sends a *single message* to Player II to let him compute the value of $f(X, Y, Z)$.
  Obviously, $[(X, Y) \rightarrow Z]$ *is at least as strong as* $[(X \leftrightarrow Y) \rightarrow Z]$.
- The bipartite models $[(X, Z) \leftrightarrow Y]$ and $[(X, Y) \rightarrow Z]$ are natural **two-party reductions** of the three-party model $[(X \leftrightarrow Y) \rightarrow Z]$.

We show that *the converse is also true* – that is, *for* ***total functions*** *the three-party model* $[(X \leftrightarrow Y) \rightarrow Z]$ *is as strong as the weaker* (with respect to the same function) *of its two-party reductions* $[(X, Z) \leftrightarrow Y]$ *and* $[(X, Y) \rightarrow Z]$.

The argument will be based on the possibility of *efficient pattern searching* in (bipartite) *PNP*.

Proof idea

For computing total functions the model $[(X \leftrightarrow Y) \rightarrow Z]$ ***is as strong as the weaker of its bipartite reductions*** $[(X, Z) \leftrightarrow Y]$ ***and*** $[(X, Y) \rightarrow Z]$**.**

Proof idea

For computing total functions the model $[(X \leftrightarrow Y) \rightarrow Z]$ ***is as strong as the weaker of its bipartite reductions*** $[(X, Z) \leftrightarrow Y]$ ***and*** $[(X, Y) \rightarrow Z]$**.**

- Assume that both $[(X, Z) \leftrightarrow Y]$ and $[(X, Y) \rightarrow Z]$ can compute $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \rightarrow \{0, 1\}$ efficiently, where $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z} \subseteq \{0, 1\}^n$.

Proof idea

For computing total functions the model $[(X \leftrightarrow Y) \rightarrow Z]$ ***is as strong as the weaker of its bipartite reductions*** $[(X, Z) \leftrightarrow Y]$ ***and*** $[(X, Y) \rightarrow Z]$**.**

- Assume that both $[(X, Z) \leftrightarrow Y]$ and $[(X, Y) \rightarrow Z]$ can compute $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \rightarrow \{0, 1\}$ efficiently, where $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z} \subseteq \{0, 1\}^n$.
- Let Π_1 be the corresponding protocol in $[(X, Y) \rightarrow Z]$ and denote by $\alpha_{x,y}$ the message send to *Player II* by *Player I* when his input is (x, y).

Proof idea

For computing total functions the model $[(X \leftrightarrow Y) \rightarrow Z]$ ***is as strong as the weaker of its bipartite reductions*** $[(X, Z) \leftrightarrow Y]$ ***and*** $[(X, Y) \rightarrow Z]$**.**

- Assume that both $[(X, Z) \leftrightarrow Y]$ and $[(X, Y) \rightarrow Z]$ can compute $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \rightarrow \{0, 1\}$ efficiently, where $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z} \subseteq \{0, 1\}^n$.
- Let Π_1 be the corresponding protocol in $[(X, Y) \rightarrow Z]$ and denote by $\alpha_{x,y}$ the message send to *Player II* by *Player I* when his input is (x, y).
- That is, $\forall\, (x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$ the message α_{x_0,y_0} allows to compute $f_{x_0,y_0}(Z) \stackrel{def}{=} f(x_0, y_0, Z)$ for every $Z \in \mathcal{Z}$.

Proof idea

For computing total functions the model $[(X \leftrightarrow Y) \rightarrow Z]$ ***is as strong as the weaker of its bipartite reductions*** $[(X, Z) \leftrightarrow Y]$ ***and*** $[(X, Y) \rightarrow Z]$**.**

- Assume that both $[(X, Z) \leftrightarrow Y]$ and $[(X, Y) \rightarrow Z]$ can compute $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \rightarrow \{0, 1\}$ efficiently, where $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z} \subseteq \{0, 1\}^n$.
- Let Π_1 be the corresponding protocol in $[(X, Y) \rightarrow Z]$ and denote by $\alpha_{x,y}$ the message send to *Player II* by *Player I* when his input is (x, y).
- That is, $\forall\, (x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$ the message α_{x_0,y_0} allows to compute $f_{x_0,y_0}(Z) \stackrel{def}{=} f(x_0, y_0, Z)$ for every $Z \in \mathcal{Z}$.
- Define $g : \mathcal{X} \times \mathcal{Y} \rightarrow \{\top, \bot\}$ as

$$g(x, y) \stackrel{\text{def}}{=} \begin{cases} \top & \text{if } \exists\, z' \in \mathcal{Z} :\ f(x, y, z') \neq 0, \\ \bot & \text{otherwise} \end{cases} = \bigvee_{z'} \begin{cases} \top & \text{if } f(x, y, z') \neq 0, \\ \bot & \text{otherwise.} \end{cases}$$

Proof idea

For computing total functions the model $[(X \leftrightarrow Y) \rightarrow Z]$ ***is as strong as the weaker of its bipartite reductions*** $[(X, Z) \leftrightarrow Y]$ ***and*** $[(X, Y) \rightarrow Z]$**.**

- Assume that both $[(X, Z) \leftrightarrow Y]$ and $[(X, Y) \rightarrow Z]$ can compute $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0, 1\}$ efficiently, where $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z} \subseteq \{0, 1\}^n$.
- Let Π_1 be the corresponding protocol in $[(X, Y) \rightarrow Z]$ and denote by $\alpha_{x,y}$ the message send to *Player II* by *Player I* when his input is (x, y).
- That is, $\forall\, (x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$ the message α_{x_0,y_0} allows to compute $f_{x_0,y_0}(Z) \stackrel{def}{=} f(x_0, y_0, Z)$ for every $Z \in \mathcal{Z}$.
- Define $g : \mathcal{X} \times \mathcal{Y} \to \{\top, \bot\}$ as

$$g(x, y) \stackrel{\text{def}}{=} \begin{cases} \top & \text{if } \exists\, z' \in \mathcal{Z} :\ f(x, y, z') \neq 0, \\ \bot & \text{otherwise} \end{cases} = \bigvee_{z'} \begin{cases} \top & \text{if } f(x, y, z') \neq 0, \\ \bot & \text{otherwise.} \end{cases}$$

- Note that $g \in PNP$:

Proof idea

For computing total functions the model $[(X \leftrightarrow Y) \rightarrow Z]$ ***is as strong as the weaker of its bipartite reductions*** $[(X, Z) \leftrightarrow Y]$ ***and*** $[(X, Y) \rightarrow Z]$.

- Assume that both $[(X, Z) \leftrightarrow Y]$ and $[(X, Y) \rightarrow Z]$ can compute $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0, 1\}$ efficiently, where $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z} \subseteq \{0, 1\}^n$.
- Let Π_1 be the corresponding protocol in $[(X, Y) \rightarrow Z]$ and denote by $\alpha_{x,y}$ the message send to *Player II* by *Player I* when his input is (x, y).
- That is, $\forall\, (x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$ the message α_{x_0,y_0} allows to compute $f_{x_0,y_0}(Z) \stackrel{def}{=} f(x_0, y_0, Z)$ for every $Z \in \mathcal{Z}$.
- Define $g : \mathcal{X} \times \mathcal{Y} \to \{\top, \bot\}$ as

$$g(x, y) \stackrel{\text{def}}{=} \begin{cases} \top & \text{if } \exists\, z' \in \mathcal{Z} :\ f(x, y, z') \neq 0, \\ \bot & \text{otherwise} \end{cases} = \bigvee_{z'} \begin{cases} \top & \text{if } f(x, y, z') \neq 0, \\ \bot & \text{otherwise.} \end{cases}$$

- Note that $g \in PNP$:
  - let the *advice* for $(x, y) \in g^{-1}(\top)$ be *any* $z' \in \mathcal{Z}$ *such that* $f(x, y, z') \neq 0$;

Proof idea

For computing total functions the model $[(X \leftrightarrow Y) \rightarrow Z]$ ***is as strong as the weaker of its bipartite reductions*** $[(X, Z) \leftrightarrow Y]$ ***and*** $[(X, Y) \rightarrow Z]$.

- Assume that both $[(X, Z) \leftrightarrow Y]$ and $[(X, Y) \rightarrow Z]$ can compute $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0, 1\}$ efficiently, where $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z} \subseteq \{0, 1\}^n$.
- Let Π_1 be the corresponding protocol in $[(X, Y) \rightarrow Z]$ and denote by $\alpha_{x,y}$ the message send to *Player II* by *Player I* when his input is (x, y).
- That is, $\forall\, (x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$ the message α_{x_0,y_0} allows to compute $f_{x_0,y_0}(Z) \stackrel{def}{=} f(x_0, y_0, Z)$ for every $Z \in \mathcal{Z}$.
- Define $g : \mathcal{X} \times \mathcal{Y} \to \{\top, \bot\}$ as

$$g(x, y) \stackrel{\text{def}}{=} \begin{cases} \top & \text{if } \exists\, z' \in \mathcal{Z} :\ f(x, y, z') \neq 0, \\ \bot & \text{otherwise} \end{cases} = \bigvee_{z'} \begin{cases} \top & \text{if } f(x, y, z') \neq 0, \\ \bot & \text{otherwise.} \end{cases}$$

- Note that $g \in PNP$:
  - let the *advice* for $(x, y) \in g^{-1}(\top)$ be *any* $z' \in \mathcal{Z}$ *such that* $f(x, y, z') \neq 0$; it can be *verified efficiently* via evaluating $f(x, y, z')$ in $[(X, Z) \leftrightarrow Y]$;

Proof idea

For computing total functions the model $[(X \leftrightarrow Y) \rightarrow Z]$ ***is as strong as the weaker of its bipartite reductions*** $[(X, Z) \leftrightarrow Y]$ ***and*** $[(X, Y) \rightarrow Z]$.

- Assume that both $[(X, Z) \leftrightarrow Y]$ and $[(X, Y) \rightarrow Z]$ can compute $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0, 1\}$ efficiently, where $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z} \subseteq \{0, 1\}^n$.
- Let Π_1 be the corresponding protocol in $[(X, Y) \rightarrow Z]$ and denote by $\alpha_{x,y}$ the message send to *Player II* by *Player I* when his input is (x, y).
- That is, $\forall\, (x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$ the message α_{x_0,y_0} allows to compute $f_{x_0,y_0}(Z) \stackrel{def}{=} f(x_0, y_0, Z)$ for every $Z \in \mathcal{Z}$.
- Define $g : \mathcal{X} \times \mathcal{Y} \to \{\top, \bot\}$ as

$$g(x, y) \stackrel{\text{def}}{=} \begin{cases} \top & \text{if } \exists\, z' \in \mathcal{Z} :\ f(x, y, z') \neq 0, \\ \bot & \text{otherwise} \end{cases} = \bigvee_{z'} \begin{cases} \top & \text{if } f(x, y, z') \neq 0, \\ \bot & \text{otherwise.} \end{cases}$$

- Note that $g \in PNP$:
  - let the *advice* for $(x, y) \in g^{-1}(\top)$ be *any* $z' \in \mathcal{Z}$ *such that* $f(x, y, z') \neq 0$; it can be *verified efficiently* via evaluating $f(x, y, z')$ in $[(X, Z) \leftrightarrow Y]$;
  - the *number of patterns* is the number of distinct $f_{x_0,y_0} \not\equiv 0$, that is, at most the number of distinct messages α_{x_0,y_0} in Π_1 (efficient by assumption).

Proof idea

For computing total functions the model $[(X \leftrightarrow Y) \rightarrow Z]$ ***is as strong as the weaker of its bipartite reductions*** $[(X, Z) \leftrightarrow Y]$ ***and*** $[(X, Y) \rightarrow Z]$.

- Assume that both $[(X, Z) \leftrightarrow Y]$ and $[(X, Y) \rightarrow Z]$ can compute $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0, 1\}$ efficiently, where $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z} \subseteq \{0, 1\}^n$.
- Let Π_1 be the corresponding protocol in $[(X, Y) \rightarrow Z]$ and denote by $\alpha_{x,y}$ the message send to *Player II* by *Player I* when his input is (x, y).
- That is, $\forall\, (x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$ the message α_{x_0,y_0} allows to compute $f_{x_0,y_0}(Z) \stackrel{def}{=} f(x_0, y_0, Z)$ for every $Z \in \mathcal{Z}$.
- Define $g : \mathcal{X} \times \mathcal{Y} \to \{\top, \bot\}$ as

$$g(x, y) \stackrel{\text{def}}{=} \begin{cases} \top & \text{if } \exists\, z' \in \mathcal{Z} :\ f(x, y, z') \neq 0, \\ \bot & \text{otherwise} \end{cases} = \bigvee_{z'} \begin{cases} \top & \text{if } f(x, y, z') \neq 0, \\ \bot & \text{otherwise.} \end{cases}$$

- Note that $g \in PNP$:
  - let the *advice* for $(x, y) \in g^{-1}(\top)$ be *any* $z' \in \mathcal{Z}$ *such that* $f(x, y, z') \neq 0$; it can be *verified efficiently* via evaluating $f(x, y, z')$ in $[(X, Z) \leftrightarrow Y]$;
  - the *number of patterns* is the number of distinct $f_{x_0,y_0} \not\equiv 0$, that is, at most the number of distinct messages α_{x_0,y_0} in Π_1 (efficient by assumption).

Cheating alert: $|\mathcal{Z}|$ can be as large as 2^n (easy to fix).

Proof idea (continued)

For computing total functions the model $[(X \leftrightarrow Y) \rightarrow Z]$ ***is as strong as the weaker of its bipartite reductions*** $[(X, Z) \leftrightarrow Y]$ ***and*** $[(X, Y) \rightarrow Z]$**.**

Proof idea (continued)

For computing total functions the model $[(X \leftrightarrow Y) \rightarrow Z]$ ***is as strong as the weaker of its bipartite reductions*** $[(X, Z) \leftrightarrow Y]$ ***and*** $[(X, Y) \rightarrow Z]$**.**

- That is, the function

$$g(x, y) = \bigvee_{z'} \begin{cases} \top & \text{if } f(x, y, z') \neq 0, \\ \bot & \text{otherwise} \end{cases}$$

  is *in PNP* (with respect to this specific decomposition).

Proof idea (continued)

For computing total functions the model $[(X \leftrightarrow Y) \rightarrow Z]$ ***is as strong as the weaker of its bipartite reductions*** $[(X, Z) \leftrightarrow Y]$ ***and*** $[(X, Y) \rightarrow Z]$**.**

- That is, the function

$$g(x, y) = \bigvee_{z'} \begin{cases} \top & \text{if } f(x, y, z') \neq 0, \\ \bot & \text{otherwise} \end{cases}$$

  is *in PNP* (with respect to this specific decomposition).

- Accordingly, for every $(x, y) \in \mathcal{X} \times \mathcal{Y}$ the corresponding *accepting pattern* – that is, the set

$$\left\{ z' \middle| f(x, y, z') \neq 0 \right\}$$

  – can be found efficiently by a deterministic bipartite protocol where Alice holds x and Bob holds y.

Proof idea (continued)

For computing total functions the model $[(X \leftrightarrow Y) \rightarrow Z]$ ***is as strong as the weaker of its bipartite reductions*** $[(X, Z) \leftrightarrow Y]$ ***and*** $[(X, Y) \rightarrow Z]$**.**

- That is, the function
$$g(x, y) = \bigvee_{z'} \begin{cases} \top & \text{if } f(x, y, z') \neq 0, \\ \bot & \text{otherwise} \end{cases}$$
is *in PNP* (with respect to this specific decomposition).
- Accordingly, for every $(x, y) \in \mathcal{X} \times \mathcal{Y}$ the corresponding *accepting pattern* – that is, the set
$$\left\{ z' \middle| f(x, y, z') \neq 0 \right\}$$
– can be found efficiently by a deterministic bipartite protocol where Alice holds x and Bob holds y.
- The number of *distinct patterns* is at most the number of distinct messages in the the protocol Π_1, which is efficient by assumption – therefore the index of the found pattern can be signalled efficiently between the players.

Proof idea (continued)

For computing total functions the model $[(X \leftrightarrow Y) \rightarrow Z]$ ***is as strong as the weaker of its bipartite reductions*** $[(X, Z) \leftrightarrow Y]$ ***and*** $[(X, Y) \rightarrow Z]$**.**

- That is, the function
$$g(x, y) = \bigvee_{z'} \begin{cases} \top & \text{if } f(x, y, z') \neq 0, \\ \bot & \text{otherwise} \end{cases}$$
is *in PNP* (with respect to this specific decomposition).
- Accordingly, for every $(x, y) \in \mathcal{X} \times \mathcal{Y}$ the corresponding *accepting pattern* – that is, the set
$$\left\{ z' \middle| f(x, y, z') \neq 0 \right\}$$
– can be found efficiently by a deterministic bipartite protocol where Alice holds x and Bob holds y.
- The number of *distinct patterns* is at most the number of distinct messages in the the protocol Π_1, which is efficient by assumption – therefore the index of the found pattern can be signalled efficiently between the players.
- The desired existence of *an efficient protocol for* $f(X, Y, Z)$ *in the model* $[(X \leftrightarrow Y) \rightarrow Z]$ follows.

Conclusions

- This work demonstrates several *qualitative equivalences among communication complexity classes*.

Conclusions

- This work demonstrates several *qualitative equivalences among communication complexity classes*.
- These results rely upon the *limitations imposed by* **total functions** *as communicational problems* (in particular, none of the equivalences hold in the case of *partial functions*).

Conclusions

- This work demonstrates several *qualitative equivalences among communication complexity classes*.
- These results rely upon the *limitations imposed by* **total functions** *as communicational problems* (in particular, none of the equivalences hold in the case of *partial functions*).
  - What other structural implications does “totality” have?

Conclusions

- This work demonstrates several *qualitative equivalences among communication complexity classes*.
- These results rely upon the *limitations imposed by* **total functions** *as communicational problems* (in particular, none of the equivalences hold in the case of *partial functions*).
  - What other structural implications does "totality" have?
  - In particular, what are the "*strengths of determinism*" and the "*limitations of non-determinism*" that are specific for the case of total functions?

Conclusions

- This work demonstrates several *qualitative equivalences among communication complexity classes*.
- These results rely upon the *limitations imposed by* **total functions** *as communicational problems* (in particular, none of the equivalences hold in the case of *partial functions*).
  - What other structural implications does "totality" have?
  - In particular, what are the "*strengths of determinism*" and the "*limitations of non-determinism*" that are specific for the case of total functions?

THANK YOU!