

Inverse protein folding for constructible structures

Arvind Gupta, Ján Maňuch, Ladislav Stacho

Simon Fraser University

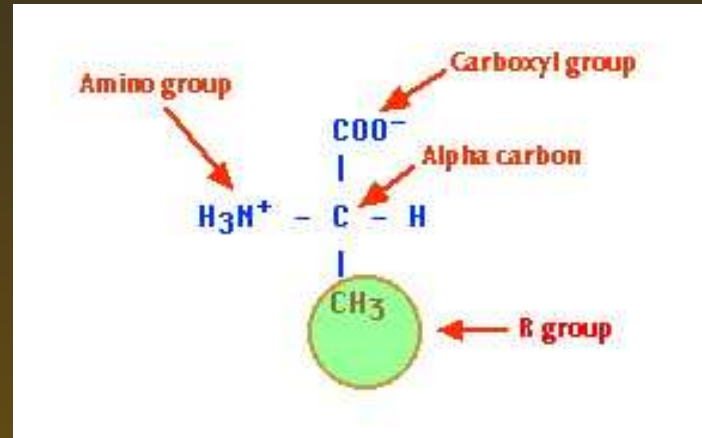
Proteins

Protein is a polymer constructed from a linear sequence (chain) of amino acids.

Proteins

Protein is a polymer constructed from a linear sequence (chain) of amino acids.

Amino acid:

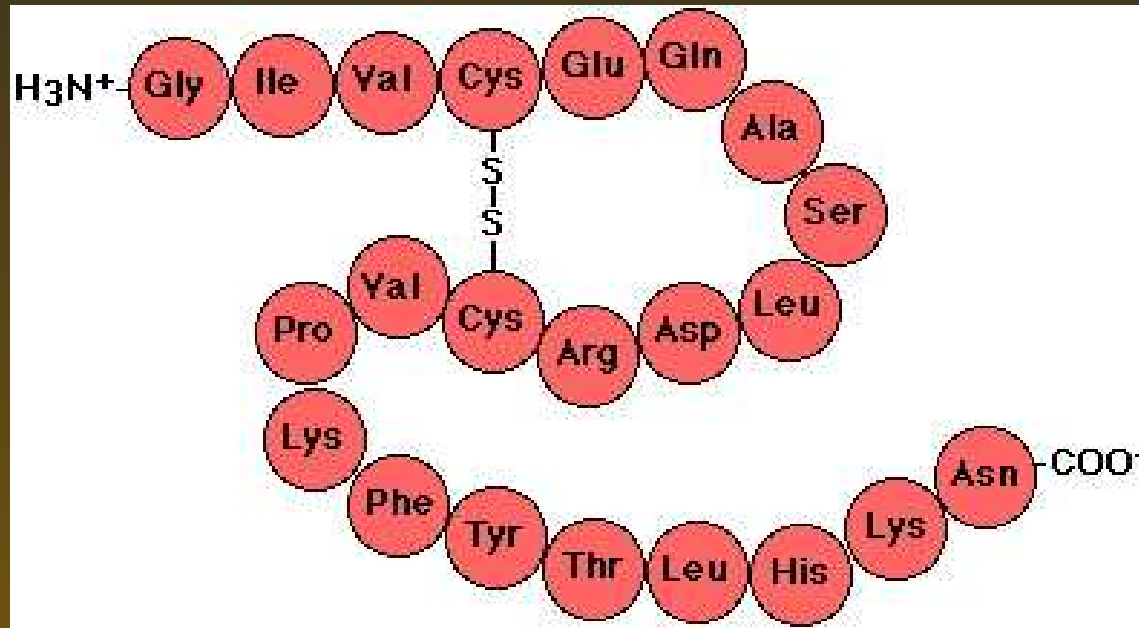


The structure of the R group determines special properties of amino acids.

There are 20 types of amino acids (Alanine, Arginine, Asparagine, Aspartic acid, Cysteine, Glutamic acid, Glutamine, Glycine, Histidine, Isoleucine, Leucine, Lysine, Methionine, Phenylalanine, Proline, Serine, Threonine, Tryptophan, Tyrosine, Valine).

Proteins

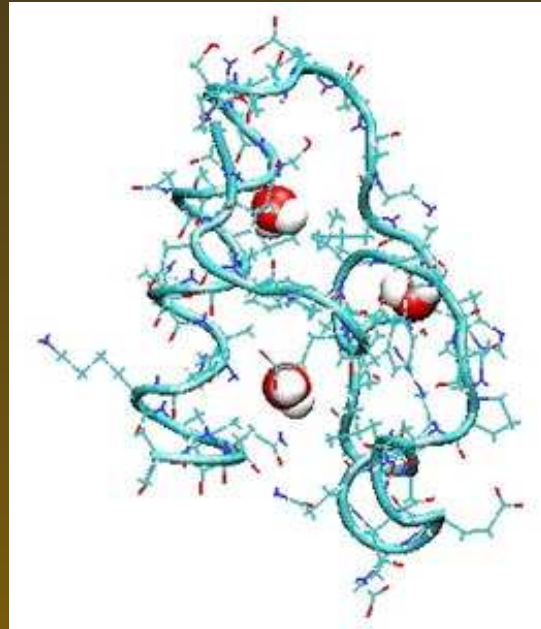
Protein is a polymer constructed from a linear sequence (chain) of amino acids.



Proteins

Protein is a polymer constructed from a linear sequence (chain) of amino acids.

When placed into a solvent it will fold into a *unique* 3D spatial structure with minimal energy.



The structure (shape) determines the *function* of the protein.

Proteins

Protein is a polymer constructed from a linear sequence (chain) of amino acids.

When placed into a solvent it will fold into a *unique* 3D spatial structure with minimal energy. The structure (shape) determines the *function* of the protein.

- It is not known how a protein can choose the minimum energy fold among all possible folds, cf. *Dill, Bromberg, Yue, Fiebig, Yee, Thomas, Chan (1995)*.

Protein Folding

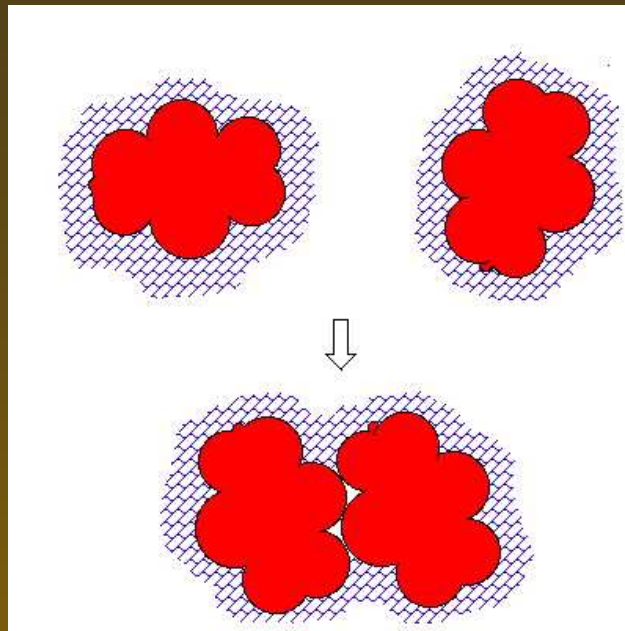
Many forces act on the protein which contribute to changes in free energy including:

- hydrogen bonding,
- van der Waals interactions,
- intrinsic propensities,
- ion pairing,
- disulphide bonds,
- hydrophobic interactions.

Protein Folding

Many forces act on the protein which contribute to changes in free energy including:

- **hydrophobic interactions** — most significant, cf. *Dill* (1990)



Protein Folding

Many forces act on the protein which contribute to changes in free energy including:

- *hydrophobic interactions* — most significant, cf. *Dill* (1990)

Amino acids are of two types: *hydrophobic* or *polar* depending on their affinity to water.

Hence, we can model proteins as sequences over $\{0, 1\}$, where

- 1 represents a hydrophobic monomer, and
- 0 represents a hydrophilic (polar) monomer.

Hydrophobic-Polar Model

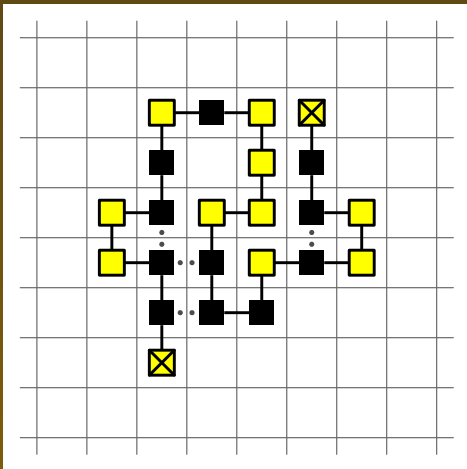
- introduced by Chan (1985)
- the protein sequence is embedded on a *2D square lattice* with each monomer occupying exactly one square and neighboring monomers occupy neighboring squares

Hydrophobic-Polar Model

- introduced by Chan (1985)
- the protein sequence is embedded on a *2D square lattice* with each monomer occupying exactly one square and neighboring monomers occupy neighboring squares

Example 1:

protein: $p_1 = 01100110100001110100110$

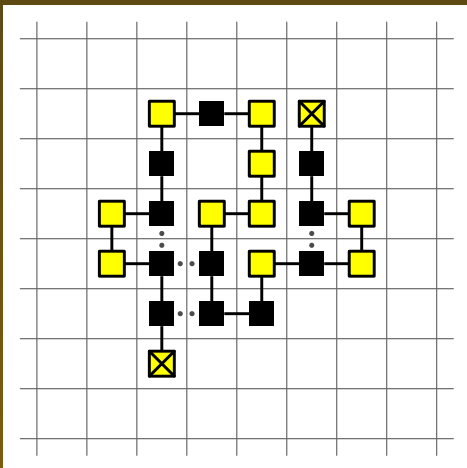


Hydrophobic-Polar Model

- introduced by Chan (1985)
- the protein sequence is embedded on a *2D square lattice* with each monomer occupying exactly one square and neighboring monomers occupy neighboring squares

Example 1:

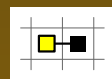
protein: $p_1 = 01100110100001110100110$ 



– depicts a polar “0” monomer



– depicts a hydrophobic “1” monomer

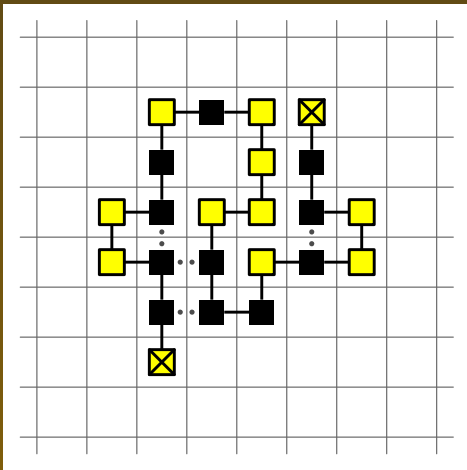


– depicts a peptide bond between neighboring monomers in the sequence

Hydrophobic-Polar Model

- introduced by Chan (1985)
- the protein sequence is embedded on a *2D square lattice* with each monomer occupying exactly one square and neighboring monomers occupy neighboring squares
- the fold of the protein is a self-avoiding walk in the lattice

Example 1:



Hydrophobic-Polar Model

- introduced by Chan (1985)
- the protein sequence is embedded on a *2D square lattice* with each monomer occupying exactly one square and neighboring monomers occupy neighboring squares
- the fold of the protein is a self-avoiding walk in the lattice
- *free energy of the fold* — in the HP model only *hydrophobic interaction* is considered: a fold with minimal free energy corresponds to a fold with the largest number of *(hydrophobic) bonds*

Hydrophobic-Polar Model

- the fold of the protein is a self-avoiding walk in the lattice
- *free energy of the fold* — in the HP model only *hydrophobic interaction* is considered: a fold with minimal free energy corresponds to a fold with the largest number of *(hydrophobic) bonds*

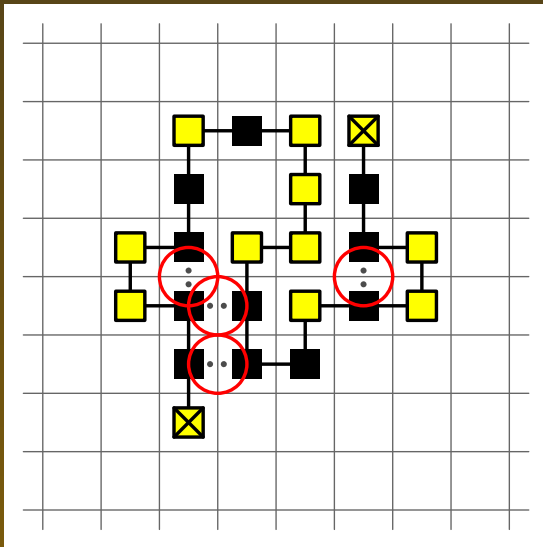
A *hydrophobic bond* occurs if we have adjacent hydrophobic “1” monomers in the lattice which are not consecutive in the protein sequence.

Hydrophobic-Polar Model

A *hydrophobic bond* occurs if we have adjacent hydrophobic “1” monomers in the lattice which are not consecutive in the protein sequence.

Example 1:

protein: $p_1 = 01100110100001110100110$



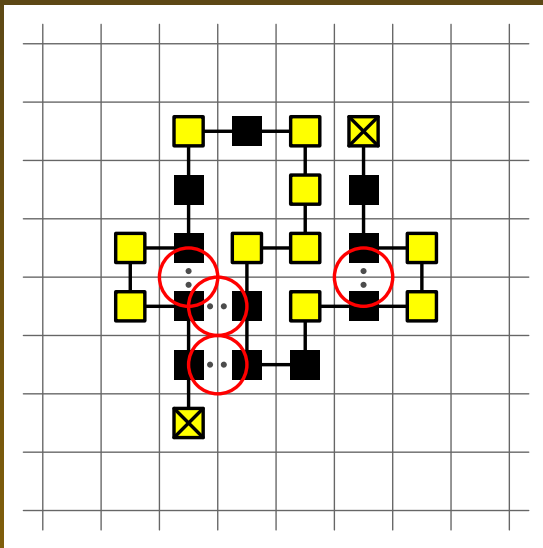
Hydrophobic-Polar Model

A *hydrophobic bond* occurs if we have adjacent hydrophobic “1” monomers in the lattice which are not consecutive in the protein sequence.

The *score* of the fold is the number of bonds.

Example 1:

protein: $p_1 = 01100110100001110100110$



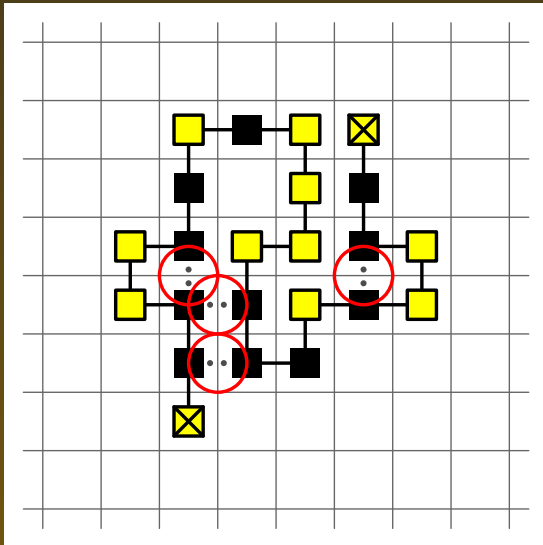
the score of this fold is 4

Hydrophobic-Polar Model

The *score* of the fold is the number of bonds.

Example 1:

protein: $p_1 = 01100110100001110100110$



the score of this fold is 4

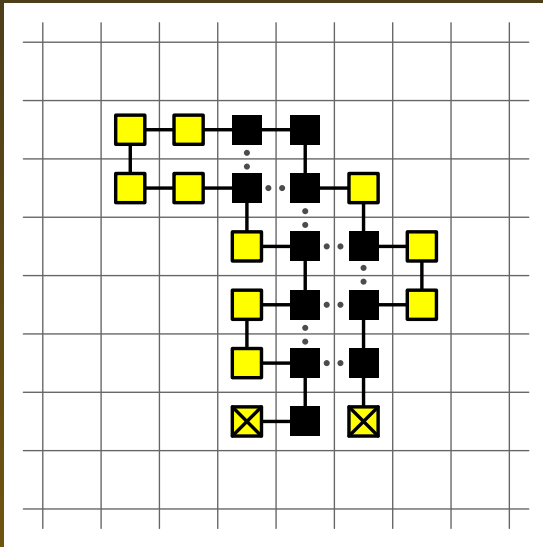
this is not the maximum score

Hydrophobic-Polar Model

The *score* of the fold is the number of bonds.

Example 1:

protein: $p_1 = 01100110100001110100110$



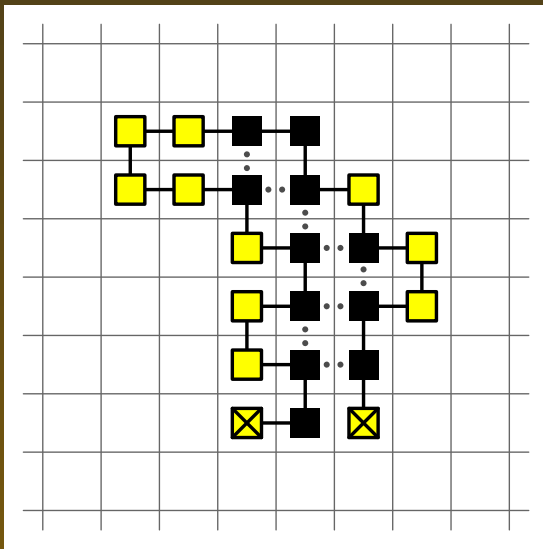
the score of this fold is 8
this is the maximum score

Hydrophobic-Polar Model

The *score* of the fold is the number of bonds.
A fold with the maximum score is called a *native* fold.

Example 1:

protein: $p_1 = 01100110100001110100110$



the score of this fold is 8

this is the maximum score

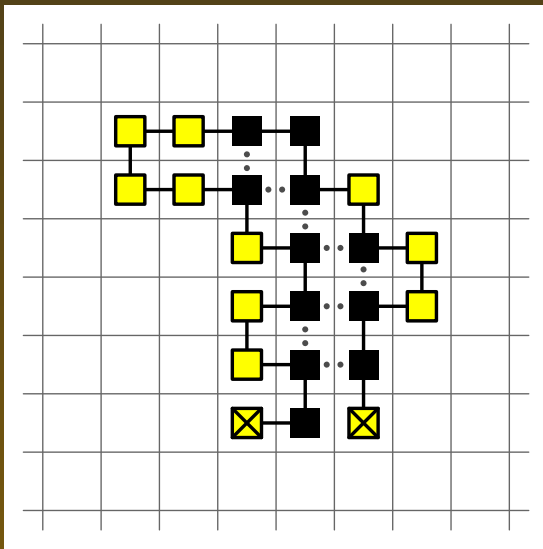
hence, this fold is a native fold

Hydrophobic-Polar Model

The *score* of the fold is the number of bonds.
A fold with the maximum score is called a *native* fold.

Example 1:

protein: $p_1 = 01100110100001110100110$



the score of this fold is 8

this is the maximum score

hence, this fold is a native fold

Protein folding problem:

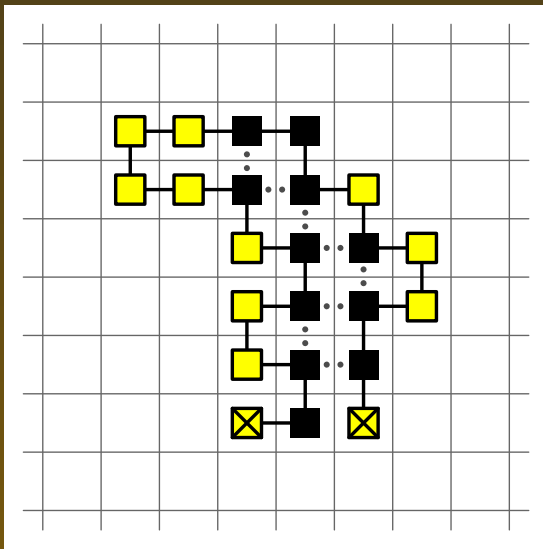
Given a protein sequence, find its native fold, cf. 

Hydrophobic-Polar Model

The *score* of the fold is the number of bonds.
A fold with the maximum score is called a *native* fold.

Example 1:

protein: $p_1 = 01100110100001110100110$



the score of this fold is 8

this is the maximum score

hence, this fold is a native fold

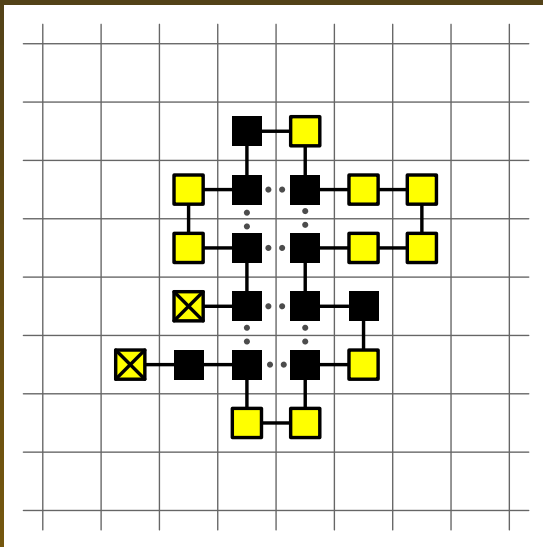
there might be several native folds
of the protein

Hydrophobic-Polar Model

The *score* of the fold is the number of bonds.
A fold with the maximum score is called a *native* fold.

Example 1:

protein: $p_1 = 01100110100001110100110$



the score of this fold is 8

this is the maximum score

hence, this fold is a native fold

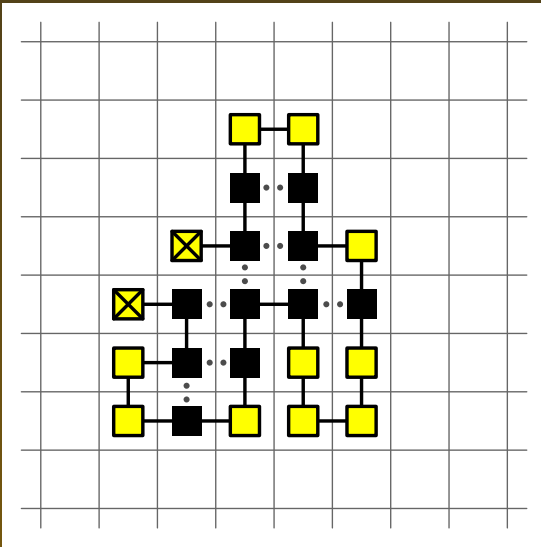
there might be several native folds
of the protein

Hydrophobic-Polar Model

The *score* of the fold is the number of bonds.
A fold with the maximum score is called a *native* fold.

Example 1:

protein: $p_1 = 01100110100001110100110$



hence, this fold is a native fold

there might be several native folds
of the protein

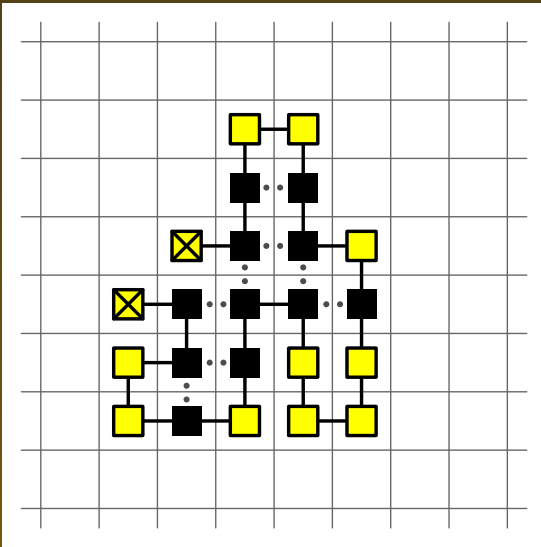
the total number of native folds of
this protein is 82

Hydrophobic-Polar Model

A fold with the maximum score is called a *native* fold.
A protein is *stable* if it has a unique native fold.

Example 1:

protein: $p_1 = 01100110100001110100110$



hence, this fold is a native fold

there might be several native folds
of the protein

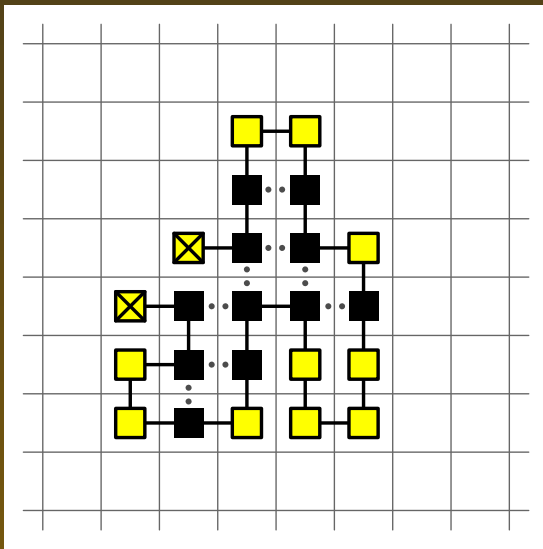
the total number of native folds of
this protein is 82

Hydrophobic-Polar Model

A fold with the maximum score is called a *native* fold.
A protein is *stable* if it has a unique native fold.

Example 1:

protein: $p_1 = 01100110100001110100110$



there might be several native folds
of the protein

the total number of native folds of
this protein is 82

hence, the protein p_1 is not stable

Hydrophobic-Polar Model

A fold with the maximum score is called a *native* fold.
A protein is *stable* if it has a unique native fold.

Example 2:

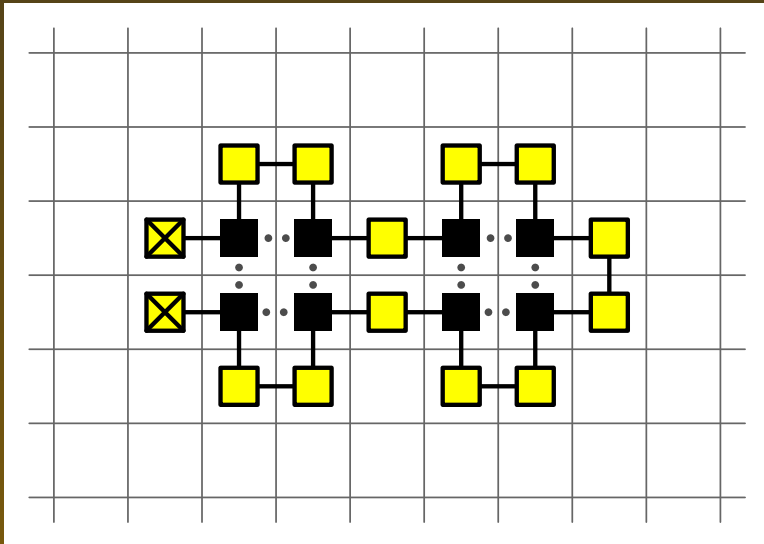
protein: $p_2 = 0100101001001001010010$

Hydrophobic-Polar Model

A fold with the maximum score is called a *native* fold.
A protein is *stable* if it has a unique native fold.

Example 2:

protein: $p_2 = 0100101001001001010010$

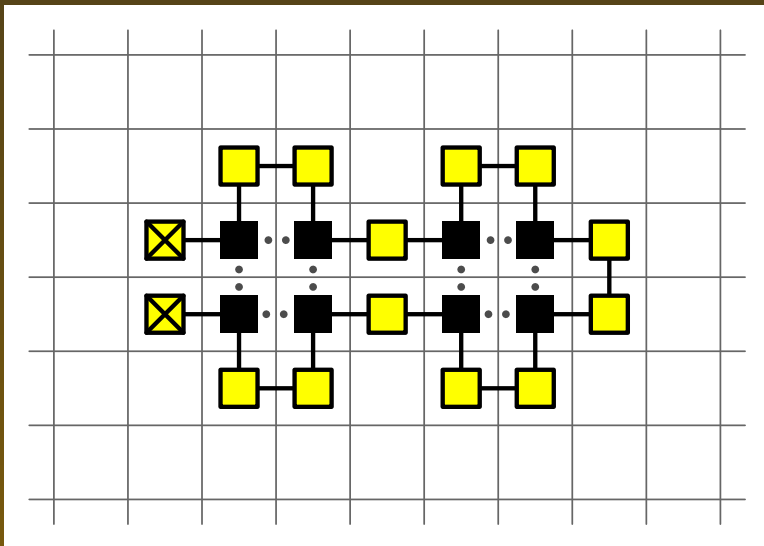


Hydrophobic-Polar Model

A fold with the maximum score is called a *native* fold.
A protein is *stable* if it has a unique native fold.

Example 2:

protein: $p_2 = 0100101001001001010010$



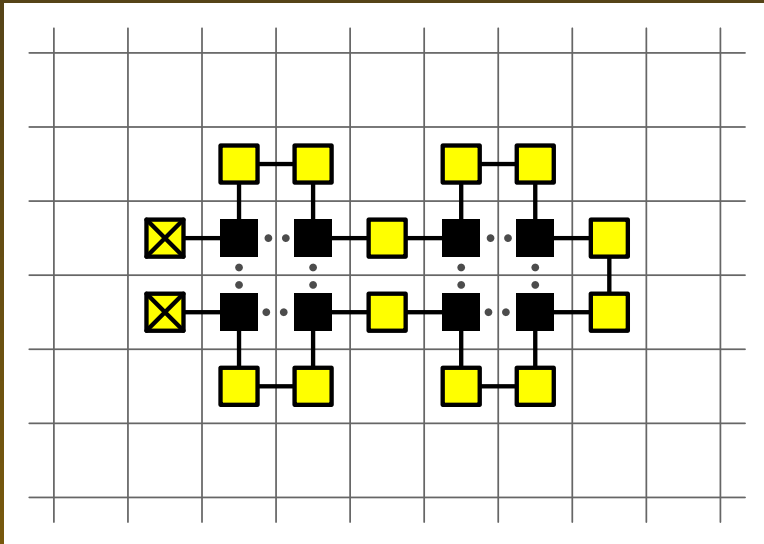
note that every hydrophobic
“1” monomer has two bonds

Hydrophobic-Polar Model

A fold with the maximum score is called a *native* fold.
A protein is *stable* if it has a unique native fold.

Example 2:

protein: $p_2 = 0100101001001001010010$



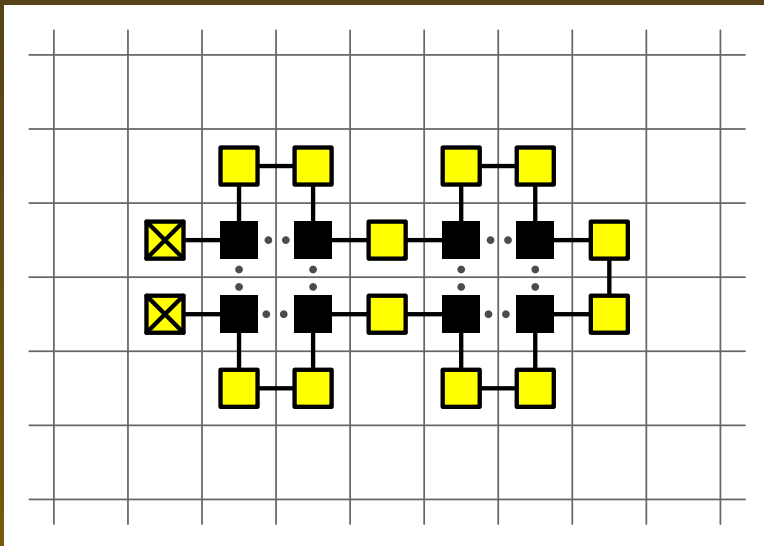
note that every hydrophobic
“1” monomer has two bonds
there is no way how to add
more bonds, i.e., this fold has
the maximum score

Hydrophobic-Polar Model

A fold with the maximum score is called a *native* fold.
A protein is *stable* if it has a unique native fold.

Example 2:

protein: $p_2 = 0100101001001001010010$



note that every hydrophobic
“1” monomer has two bonds

there is no way how to add
more bonds, i.e., this fold has
the maximum score

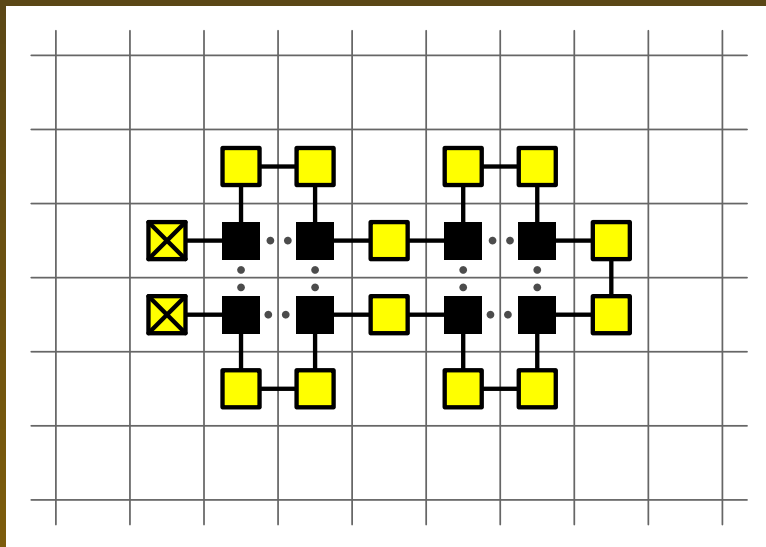
hence, this is a native fold of
the protein p_2

Hydrophobic-Polar Model

A fold in which every hydrophobic “1” monomer has the maximal number of bonds is called *saturated*.

Example 2:

protein: $p_2 = 0100101001001001010010$



note that every hydrophobic “1” monomer has two bonds
there is no way how to add more bonds, i.e., this fold has the maximum score

hence, this is a native fold of the protein p_2

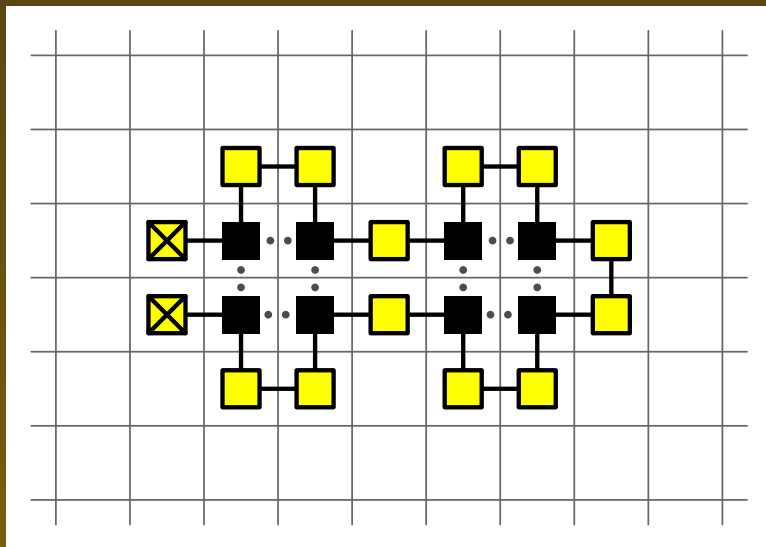
Hydrophobic-Polar Model

Assume that the protein is from the set $0\{0,1\}^*0$.

A fold in which every hydrophobic “1” monomer has two bonds is called *saturated*.

Example 2:

protein: $p_2 = 0100101001001001010010$



note that every hydrophobic “1” monomer has two bonds

there is no way how to add more bonds, i.e., this fold has the maximum score

hence, this is a native fold of the protein p_2

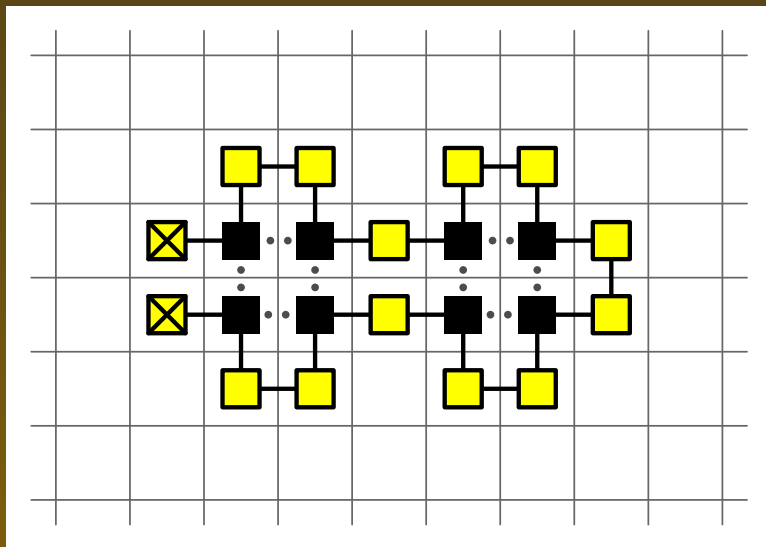
Hydrophobic-Polar Model

Assume that the protein is from the set $0\{0,1\}^*0$.

A fold in which every hydrophobic “1” monomer has two bonds is called *saturated*.

Example 2:

protein: $p_2 = 0100101001001001010010$



there is no way how to add more bonds, i.e., this fold has the maximum score

hence, this is a native fold of the protein p_2

this fold is saturated

Hydrophobic-Polar Model

Assume that the protein is from the set $0\{0,1\}^*0$.

A fold in which every hydrophobic “1” monomer has two bonds is called *saturated*.

Observation 1: *If there exists a saturated fold of a protein p then the fold is native, and any native fold of p is saturated.*

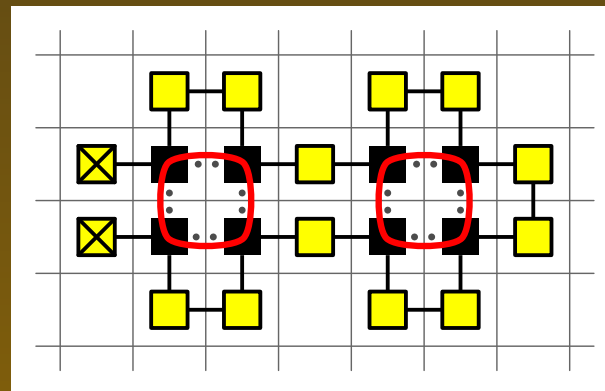
Hydrophobic-Polar Model

Assume that the protein is from the set $0\{0,1\}^*0$.

A fold in which every hydrophobic “1” monomer has two bonds is called *saturated*.

Observation 1: *If there exists a saturated fold of a protein p then the fold is native, and any native fold of p is saturated.*

Note that in a saturated fold hydrophobic “1” monomers form cycles (called *1-cycles*) where all edges are bonds.

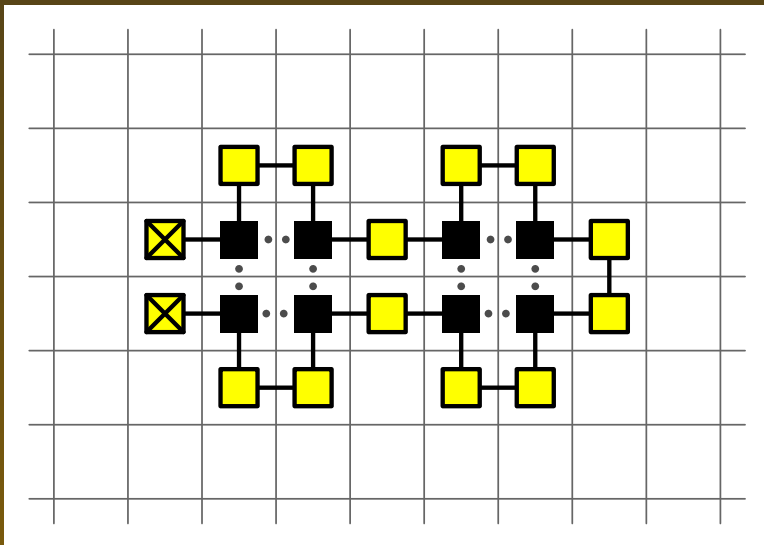


Hydrophobic-Polar Model

Observation 1: *If there exists a saturated fold of a protein p then the fold is native, and any native fold of p is saturated.*

Example 2:

protein: $p_2 = 0100101001001001010010$



by Observation, every native fold of p_2 is saturated

Inverse protein folding

In many applications such as *drug design*, we are actually interested in the complement problem to protein folding:

Inverse protein folding

In many applications such as *drug design*, we are actually interested in the complement problem to protein folding:

For a target fold, find a protein sequence whose native fold is the target.

Inverse protein folding

In many applications such as *drug design*, we are actually interested in the complement problem to protein folding:

For a target fold, find a protein sequence whose native fold is the target.

Early work on this problem involved:

- heuristics that buried the H monomers in a central core with the P monomers on the outside, cf. *Kamtekar, Schiffer, Xiong, Babik, Hecht (1993)*

Inverse protein folding

In many applications such as *drug design*, we are actually interested in the complement problem to protein folding:

For a target fold, find a protein sequence whose native fold is the target.

Early work on this problem involved:

- heuristics that buried the H monomers in a central core with the P monomers on the outside, cf. *Kamtekar, Schiffer, Xiong, Babik, Hecht (1993)*
- finding all possible short sequences and putting these together, cf. *Yue, Dill (1992)*

Inverse protein folding

In many applications such as *drug design*, we are actually interested in the complement problem to protein folding:

For a target fold, find a protein sequence whose native fold is the target.

Early work on this problem involved:

- heuristics that buried the H monomers in a central core with the P monomers on the outside, cf. *Kamtekar, Schiffer, Xiong, Babik, Hecht (1993)*
- finding all possible short sequences and putting these together, cf. *Yue, Dill (1992)*
- sequence evolution, a form of local search, cf. *Sun, Brem, Chan, Dill (1995)*

Inverse protein folding

More natural formulation of the *inverse protein folding problem*:

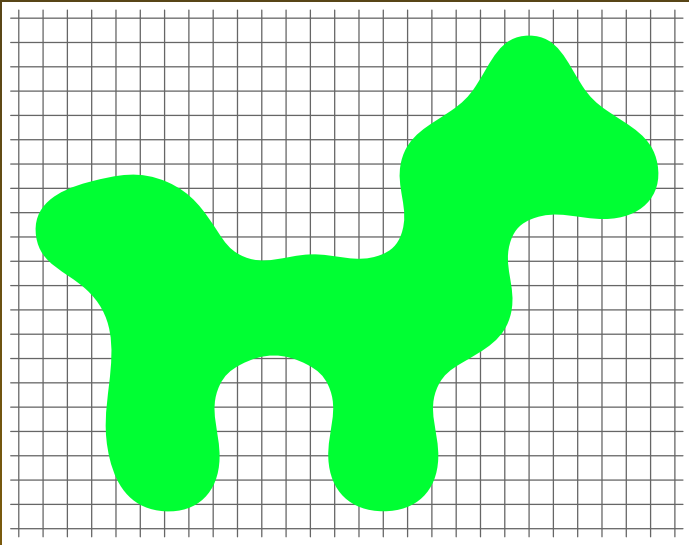
For a given shape find a protein with a native fold approximating the shape.

Inverse protein folding

More natural formulation of the *inverse protein folding problem*:

For a given shape find a protein with a native fold approximating the shape.

Example:



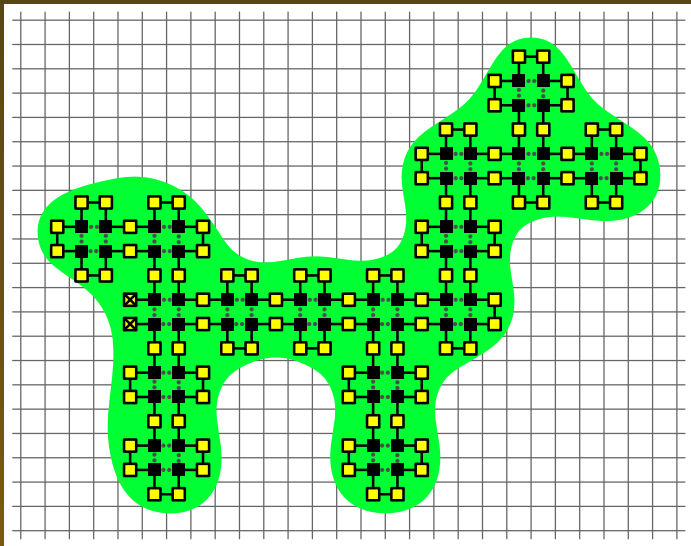
We are given a **target shape**.

Inverse protein folding

More natural formulation of the *inverse protein folding problem*:

For a given shape find a protein with a native fold approximating the shape.

Example:



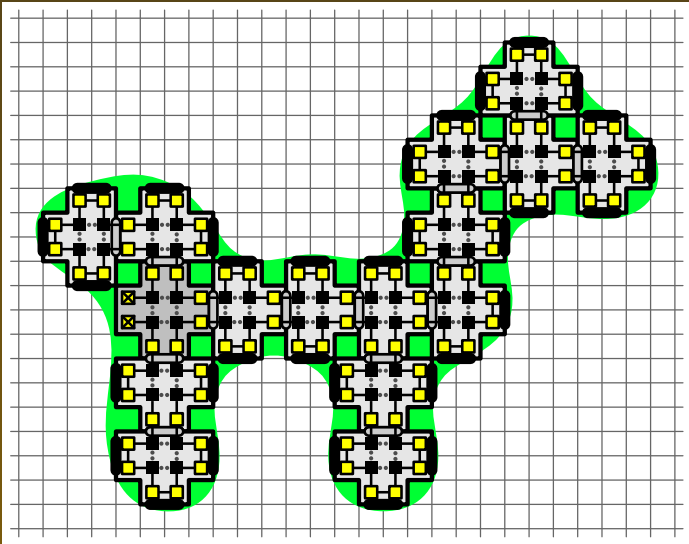
We are able to fill it with a chain monomers such that almost all squares covered by the target shape are occupied by the monomers of the chain.

Inverse protein folding

More natural formulation of the *inverse protein folding problem*:

For a given shape find a protein with a native fold approximating the shape.

Example:



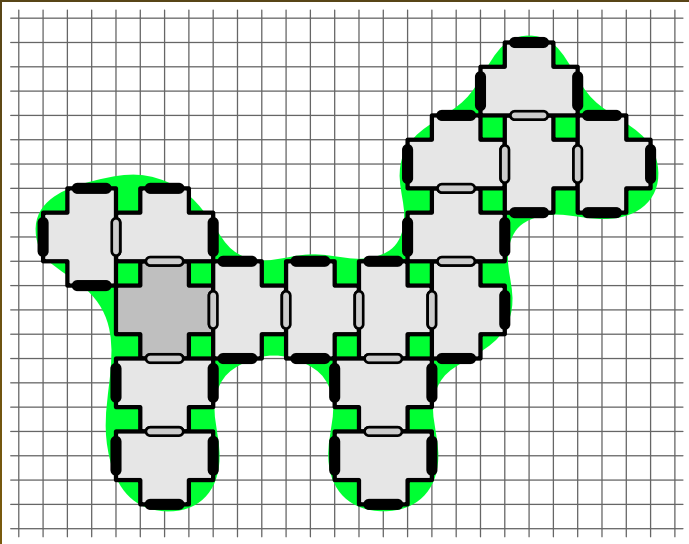
The shape is approximated by a system of tiles.

Inverse protein folding

More natural formulation of the *inverse protein folding problem*:

For a given shape find a protein with a native fold approximating the shape.

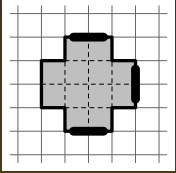
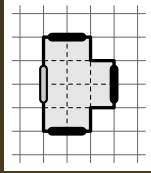
Example:



This system of tiles will be called a *constructible structure*.

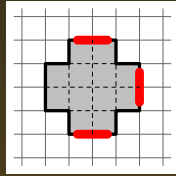
Constructible structures

We have two tiles:

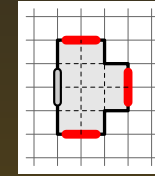
(a) a *starting* tile: , and (b) a *regular* tile: 

Constructible structures

We have two tiles:



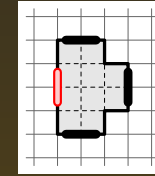
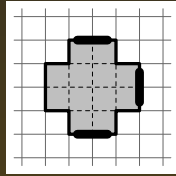
(a) a *starting* tile: , and (b) a *regular* tile:



Both tiles have three *ligands*,

Constructible structures

We have two tiles:

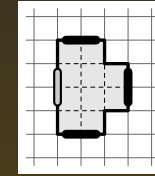
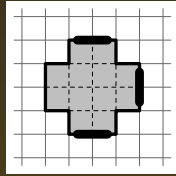


(a) a *starting* tile: , and (b) a *regular* tile: 

Both tiles have three *ligands*, and in addition, the regular tile has one *receptor*.

Constructible structures

We have two tiles:



(a) a *starting* tile: , and (b) a *regular* tile: 

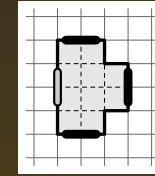
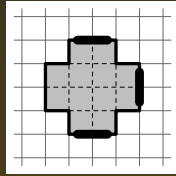
Both tiles have three *ligands*, and in addition, the regular tile has one *receptor*.

A *constructible structure* is a partial tiling of the lattice obtained by the following procedure:

1. Place the starting tile into the grid.

Constructible structures

We have two tiles:



(a) a *starting* tile: , and (b) a *regular* tile: 

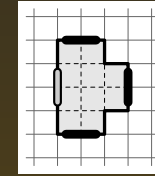
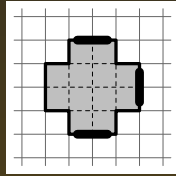
Both tiles have three *ligands*, and in addition, the regular tile has one *receptor*.

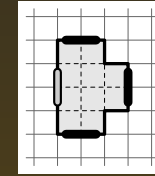
A *constructible structure* is a partial tiling of the lattice obtained by the following procedure:

1. Place the starting tile into the grid.
2. Place a regular tile into the grid so that its receptor is attached to a ligand of a tile already in the grid and it does not overlap with any other tile.

Constructible structures


We have two tiles:



(a) a *starting* tile: , and (b) a *regular* tile: 

Both tiles have three *ligands*, and in addition, the regular tile has one *receptor*.

A *constructible structure* is a partial tiling of the lattice obtained by the following procedure:

1. Place the starting tile into the grid.
2. Place a regular tile into the grid so that its receptor is attached to a ligand of a tile already in the grid and it does not overlap with any other tile.
3. Continue with step 2., or end the procedure. 

Compatible folds

A fold is *compatible with* a constructible structure S , if it covers exactly the squares of S .

Compatible folds

A fold is *compatible with* a constructible structure S , if it covers exactly the squares of S .

Lemma 1: *For every constructible structure S , there exists a fold compatible with S .*

Compatible folds

A fold is *compatible with* a constructible structure S , if it covers exactly the squares of S .

Lemma 1: *For every constructible structure S , there exists a fold compatible with S .*

Proof: By induction on the number k of regular tiles.

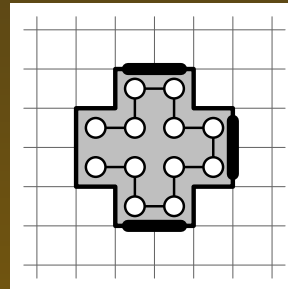
Compatible folds

A fold is *compatible with* a constructible structure S , if it covers exactly the squares of S .

Lemma 1: *For every constructible structure S , there exists a fold compatible with S .*

Proof: By induction on the number k of regular tiles.

Base case: $k = 0$



Compatible folds

A fold is *compatible with* a constructible structure S , if it covers exactly the squares of S .

Lemma 1: *For every constructible structure S , there exists a fold compatible with S .*

Proof: By induction on the number k of regular tiles.

Inductive step: Let S' be a constructive shape obtained from S by removing one regular tile.

Compatible folds

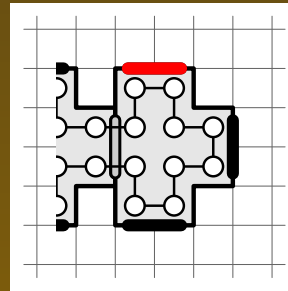
A fold is *compatible with* a constructible structure S , if it covers exactly the squares of S .

Lemma 1: *For every constructible structure S , there exists a fold compatible with S .*

Proof: By induction on the number k of regular tiles.

Inductive step: Let S' be a constructive shape obtained from S by removing one regular tile.

By the induction hyp. we have a fold compatible with S' :



Compatible folds

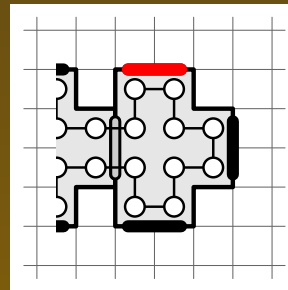
A fold is *compatible with* a constructible structure S , if it covers exactly the squares of S .

Lemma 1: *For every constructible structure S , there exists a fold compatible with S .*

Proof: By induction on the number k of regular tiles.

Inductive step: Let S' be a constructive shape obtained from S by removing one regular tile.

By the induction hyp. we have a fold compatible with S' :



Red ligand marks the place where the regular tile was cut.

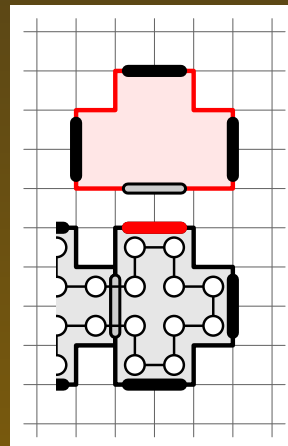
Compatible folds

Lemma 1: *For every constructible structure S , there exists a fold compatible with S .*

Proof: By induction on the number k of regular tiles.

Inductive step:

Let us attach the cut regular tile (red) back to its place.



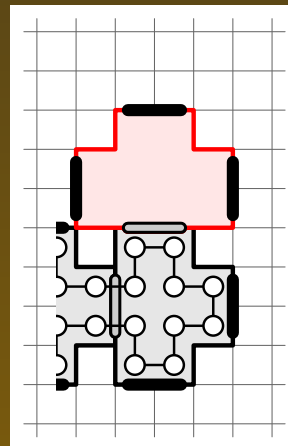
Compatible folds

Lemma 1: *For every constructible structure S , there exists a fold compatible with S .*

Proof: By induction on the number k of regular tiles.

Inductive step:

Disconnect the folding path.



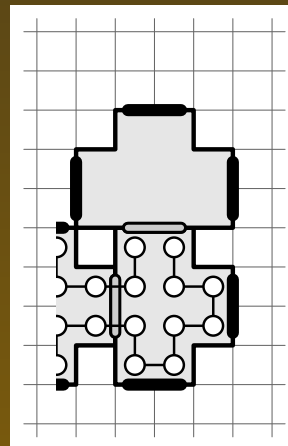
Compatible folds

Lemma 1: *For every constructible structure S , there exists a fold compatible with S .*

Proof: By induction on the number k of regular tiles.

Inductive step:

Fill the attached tile.



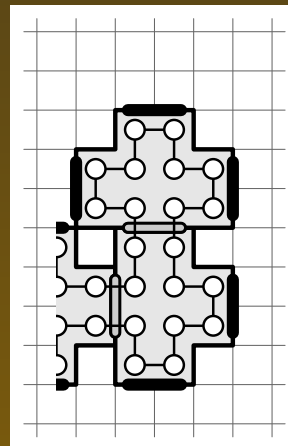
Compatible folds

Lemma 1: *For every constructible structure S , there exists a fold compatible with S .*

Proof: By induction on the number k of regular tiles.

Inductive step:

Done.



Compatible folds

Lemma 1: *For every constructible structure S , there exists a fold compatible with S .*

Proof: By induction on the number k of regular tiles.

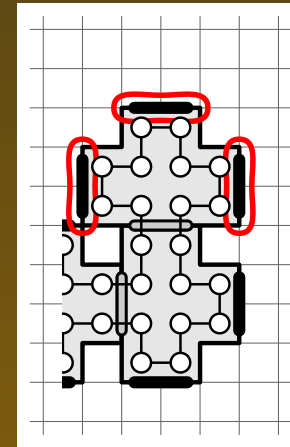
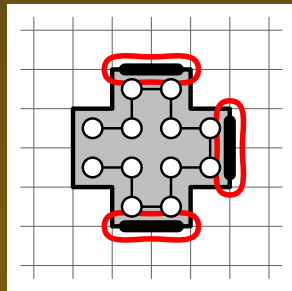
Note that the induction works because in each step: every “free” ligand is adjacent to two monomers of S which are connected with a peptide bond.

Compatible folds

Lemma 1: *For every constructible structure S , there exists a fold compatible with S .*

Proof: By induction on the number k of regular tiles.

Note that the induction works because in each step: every “free” ligand is adjacent to two monomers of S which are connected with a peptide bond.



Compatible folds

Lemma 1: *For every constructible structure S , there exists a fold compatible with S .*

Denote the fold (path) constructed in Lemma 1 by $c(S)$.

Construction of proteins

Theorem 1: *For any constructible structure S , there exists a protein $p(S)$ such that the fold $c(S)$ is saturated. Hence, $c(S)$ is a native fold of $p(S)$ and any native fold of $p(S)$ is saturated.*

Construction of proteins

Theorem 1: *For any constructible structure S , there exists a protein $p(S)$ such that the fold $c(S)$ is saturated. Hence, $c(S)$ is a native fold of $p(S)$ and any native fold of $p(S)$ is saturated.*

Proof: By induction on the number k of regular tiles.

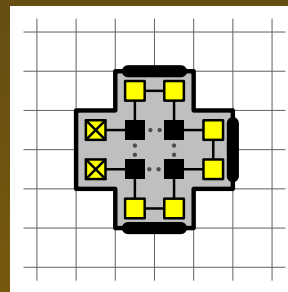
Construction of proteins

Theorem 1: *For any constructible structure S , there exists a protein $p(S)$ such that the fold $c(S)$ is saturated. Hence, $c(S)$ is a native fold of $p(S)$ and any native fold of $p(S)$ is saturated.*

Proof: By induction on the number k of regular tiles.

Base case: $k = 0$

set $p(S) = 010010010010$

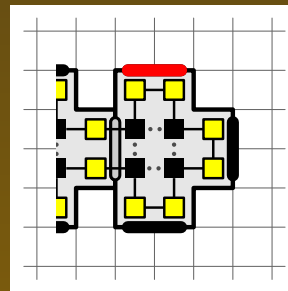


Construction of proteins

Theorem 1: *For any constructible structure S , there exists a protein $p(S)$ such that the fold $c(S)$ is saturated. Hence, $c(S)$ is a native fold of $p(S)$ and any native fold of $p(S)$ is saturated.*

Inductive step: Let S' be a constructive shape obtained from S by removing one regular tile.

By the induction hypothesis we have a protein $p(S')$ such that the fold $c(S')$ is saturated:



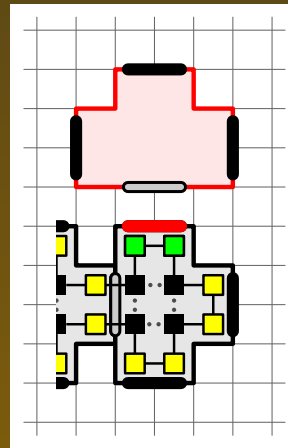
Red ligand marks the place where the regular tile was cut.

Construction of proteins

Theorem 1: *For any constructible structure S , there exists a protein $p(S)$ such that the fold $c(S)$ is saturated. Hence, $c(S)$ is a native fold of $p(S)$ and any native fold of $p(S)$ is saturated.*

Inductive step:

Attach the cut regular tile (red) back to its place.



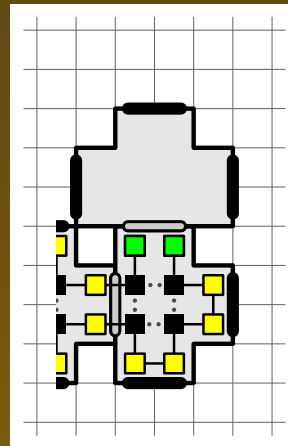
$$p(S') = p_1 \dots p_{i-1} \mathbf{00} p_{i+2} \dots p_{2+10k}$$

Construction of proteins

Theorem 1: *For any constructible structure S , there exists a protein $p(S)$ such that the fold $c(S)$ is saturated. Hence, $c(S)$ is a native fold of $p(S)$ and any native fold of $p(S)$ is saturated.*

Inductive step:

Fill the attached tile.



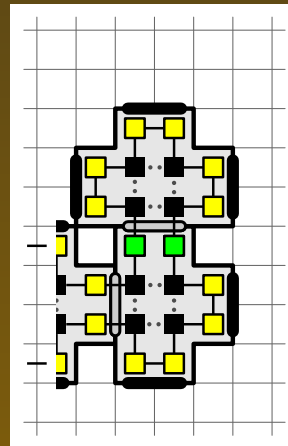
$$p_1 \dots p_{i-1} 0 \quad 0 p_{i+2} \dots p_{2+10k}$$

Construction of proteins

Theorem 1: *For any constructible structure S , there exists a protein $p(S)$ such that the fold $c(S)$ is saturated. Hence, $c(S)$ is a native fold of $p(S)$ and any native fold of $p(S)$ is saturated.*

Inductive step:

Done.



$$p(S) = p_1 \dots p_{i-1} \mathbf{010010010010} p_{i+2} \dots p_{2+10k}$$

Construction of proteins

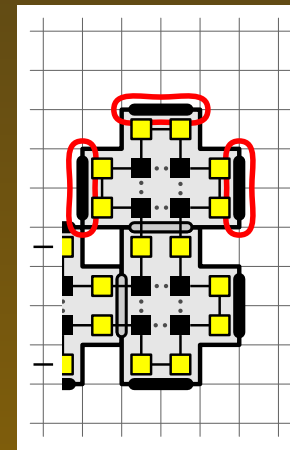
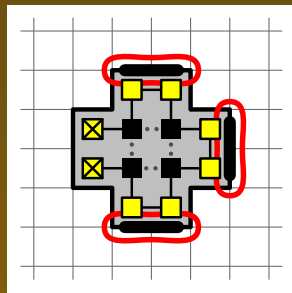
Theorem 1: *For any constructible structure S , there exists a protein $p(S)$ such that the fold $c(S)$ is saturated. Hence, $c(S)$ is a native fold of $p(S)$ and any native fold of $p(S)$ is saturated.*

Note that the induction works because in each step: every “free” ligand is adjacent to two polar “0” monomers of S which are connected with a peptide bond.

Construction of proteins

Theorem 1: *For any constructible structure S , there exists a protein $p(S)$ such that the fold $c(S)$ is saturated. Hence, $c(S)$ is a native fold of $p(S)$ and any native fold of $p(S)$ is saturated.*

Note that the induction works because in each step: every “free” ligand is adjacent to two polar “0” monomers of S which are connected with a peptide bond.



Construction of proteins

Theorem 1: *For any constructible structure S , there exists a protein $p(S)$ such that the fold $c(S)$ is saturated. Hence, $c(S)$ is a native fold of $p(S)$ and any native fold of $p(S)$ is saturated.*

The second part of the theorem follows by Observation 1.

Construction of proteins

Theorem 1: *For any constructible structure S , there exists a protein $p(S)$ such that the fold $c(S)$ is saturated. Hence, $c(S)$ is a native fold of $p(S)$ and any native fold of $p(S)$ is saturated.*

We have solved **the problem**.

However, it's desirable that the protein with native fold approximating the shape is *stable*.

Construction of proteins

Theorem 1: *For any constructible structure S , there exists a protein $p(S)$ such that the fold $c(S)$ is saturated. Hence, $c(S)$ is a native fold of $p(S)$ and any native fold of $p(S)$ is saturated.*

We have solved **the problem**.

However, it's desirable that the protein with native fold approximating the shape is *stable*.

Conjecture 1: *For any constructible structure S , the protein $p(S)$ is stable.*

Construction of proteins

Theorem 1: *For any constructible structure S , there exists a protein $p(S)$ such that the fold $c(S)$ is saturated. Hence, $c(S)$ is a native fold of $p(S)$ and any native fold of $p(S)$ is saturated.*

We have solved **the problem**.

However, it's desirable that the protein with native fold approximating the shape is *stable*.

Conjecture 1: *For any constructible structure S , the protein $p(S)$ is stable.*

The conjecture has been tested for over 16,000 constructible structures (including all structured with 0 up to 7 regular tiles).

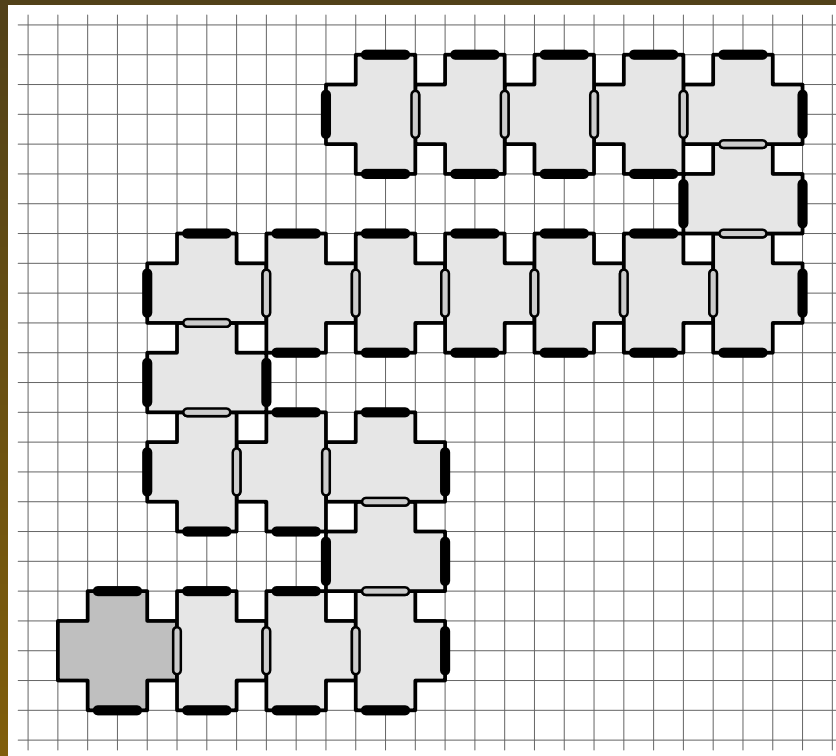
Linear constructible structures

We say that a constructible structure S is *linear* if it is constructed such that every regular tile is attached to the ligand of the *last* placed tile.

Linear constructible structures

We say that a constructible structure S is *linear* if it is constructed such that every regular tile is attached to the ligand of the *last* placed tile.

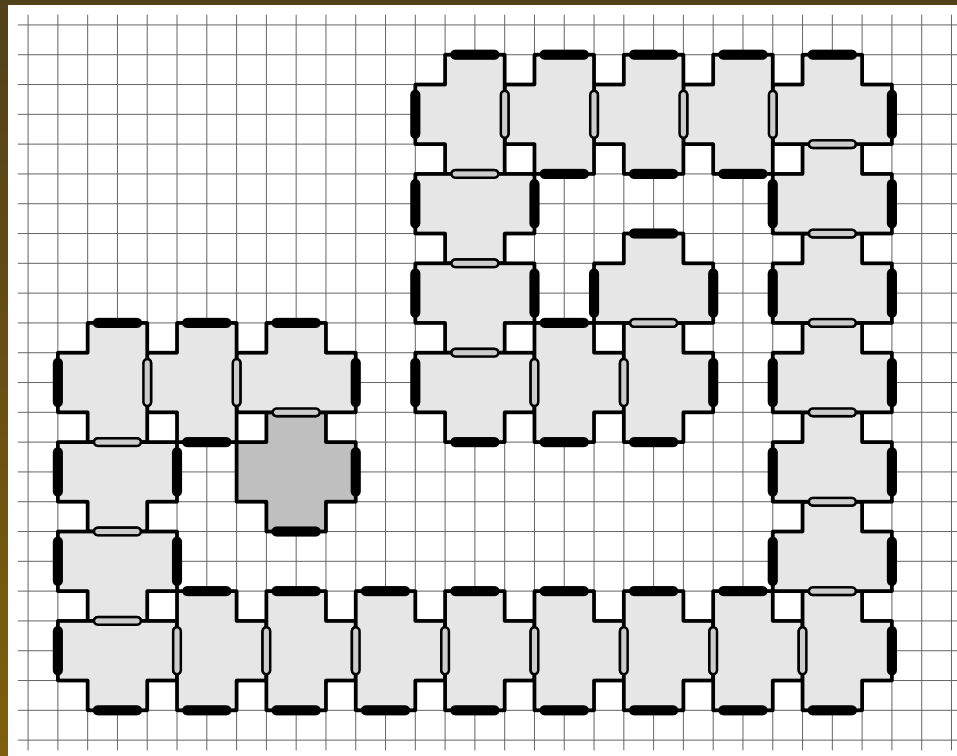
Examples:



Linear constructible structures

We say that a constructible structure S is *linear* if it is constructed such that every regular tile is attached to the ligand of the *last* placed tile.

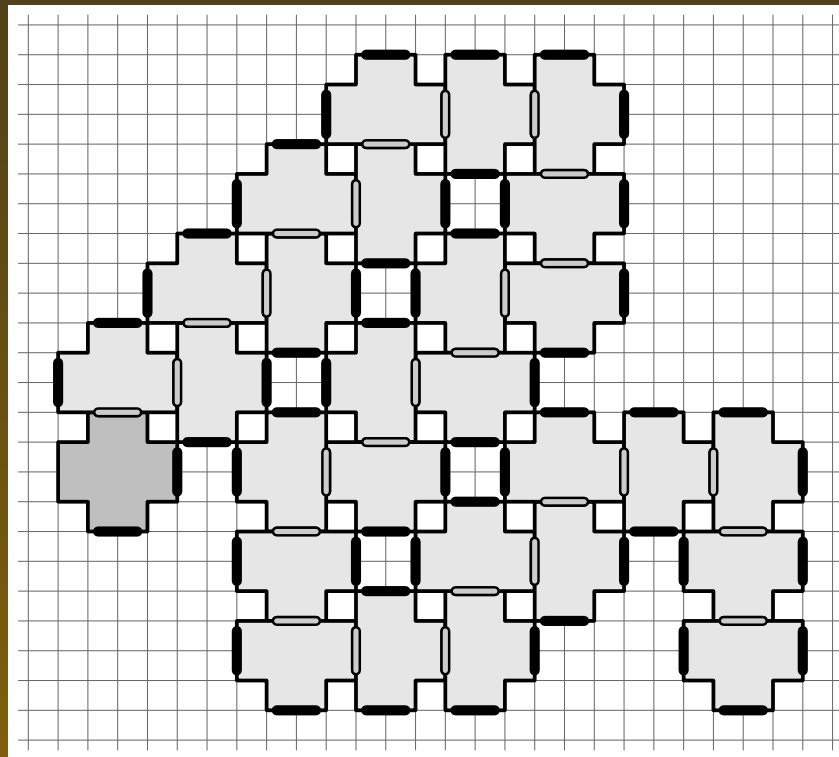
Examples:



Linear constructible structures

We say that a constructible structure S is *linear* if it is constructed such that every regular tile is attached to the ligand of the *last* placed tile.

Examples:



Linear constructible structures

We say that a constructible structure S is *linear* if it is constructed such that every regular tile is attached to the ligand of the *last* placed tile.

Conjecture 2: *For any linear constructible structure S , the protein $p(S)$ is stable.*

Linear constructible structures

We say that a constructible structure S is *linear* if it is constructed such that every regular tile is attached to the ligand of the *last* placed tile.

Description.

A linear constructible structures can be described by a *tiling sequence* in $\{1, 2, 3\}^n$ where, informally,

- the number 1 in this tiling sequence means “*turn right*”,
- the number 2 means “*continue straight*”, and
- the number 3 means “*turn left*”

when traveling along the linear chain of tiles.

Example: 

Linear constructible structures

Classification of linear constructible structures.

We can factorize the set of all linear constructible structures into classes by the number of “*bends*”, i.e., the number of 1’s and 3’s in the tiling sequence:

$$\mathcal{L}_n = \{ \text{all linear constructible structures with } n \text{ bends} \}$$

A linear constructible structure with n bends is called \mathcal{L}_n -*structure*.

Linear constructible structures

Classification of linear constructible structures.

We can factorize the set of all linear constructible structures into classes by the number of “*bends*”, i.e., the number of 1’s and 3’s in the tiling sequence:

$$\mathcal{L}_n = \{ \text{all linear constructible structures with } n \text{ bends} \}$$

A linear constructible structure with n bends is called *\mathcal{L}_n -structure*.

Our results:

If S is \mathcal{L}_0 -structure or \mathcal{L}_1 -structure then $p(S)$ is stable.

Properties of $p(S)$

Observation 2: *For any constructible structure S , the protein $p(S)$ satisfies the following properties:*

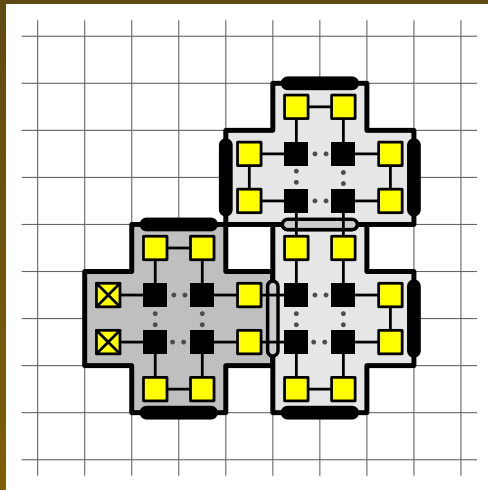
- $p(S) \in 0\{0, 1\}^*0$, and
- $p(S)$ does not contain any of 11, 000, 1010101 and $100100100100 = (100)^4$ as a substring.

Properties of $p(S)$

Observation 2: *For any constructible structure S , the protein $p(S)$ satisfies the following properties:*

- $p(S) \in 0\{0,1\}^*0$, and
- $p(S)$ does not contain any of 11 , 000 , 1010101 and $100100100100 = (100)^4$ as a substring.

Example:

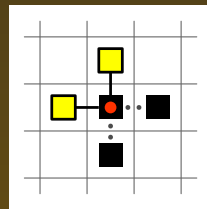


$$p_3 = 0100101001001010010010010010101001$$

Properties of $p(S)$

Observation 3: *Every saturated fold of $p(S)$ has the following properties:*

(a) *every 1-monomer has two 1-monomers and two 0-monomers as neighbors;*



Properties of $p(S)$

Observation 3: *Every saturated fold of $p(S)$ has the following properties:*

- (a) *every 1-monomer has two 1-monomers and two 0-monomers as neighbors;*
- (b) *every 0-monomer has at least one adjacent 1-monomer;*
(since 000 is not a substring of $p(S)$)

Properties of $p(S)$

Observation 3: *Every saturated fold of $p(S)$ has the following properties:*

- (a) every 1-monomer has two 1-monomers and two 0-monomers as neighbors;*
- (b) every 0-monomer has at least one adjacent 1-monomer;*
- (c) an adjacent 1-monomer and 0-monomer are connected with a peptide bond;*

(follows by (a))

Properties of $p(S)$

Observation 3: *Every saturated fold of $p(S)$ has the following properties:*

- (a) every 1-monomer has two 1-monomers and two 0-monomers as neighbors;*
- (b) every 0-monomer has at least one adjacent 1-monomer;*
- (c) an adjacent 1-monomer and 0-monomer are connected with a peptide bond;*
- (d) adjacent 1-monomers are connected by a bond.*

(since the fold is saturated)

Properties of $p(S)$

Observation 3: *Every saturated fold of $p(S)$ has the following properties:*

- (a) every 1-monomer has two 1-monomers and two 0-monomers as neighbors;*
- (b) every 0-monomer has at least one adjacent 1-monomer;*
- (c) an adjacent 1-monomer and 0-monomer are connected with a peptide bond;*
- (d) adjacent 1-monomers are connected by a bond.*

\mathcal{L}_0 -structures

\mathcal{L}_0 -structures:

- ▣ their tiling sequences can contain only 2's (“go straight”)

\mathcal{L}_0 -structures

\mathcal{L}_0 -structures:

- their tiling sequences can contain only 2's (“go straight”)
- let S_n be a linear constructible structure described by the sequence: $\underbrace{2, 2, \dots, 2}_{n-1}$

\mathcal{L}_0 -structures

\mathcal{L}_0 -structures:

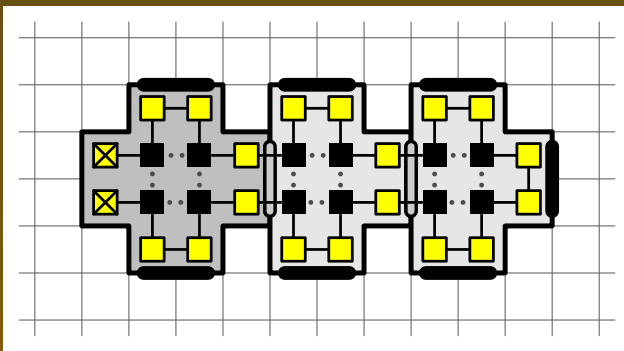
- their tiling sequences can contain only 2's ("go straight")
- let S_n be a linear constructible structure described by the sequence: $\underbrace{2, 2, \dots, 2}_{n-1}$
- we have $p(S_n) = 0(10010)^n(01001)^n0$

\mathcal{L}_0 -structures

\mathcal{L}_0 -structures:

- their tiling sequences can contain only 2's ("go straight")
- let S_n be a linear constructible structure described by the sequence: $\underbrace{2, 2, \dots, 2}_{n-1}$
- we have $p(S_n) = 0(10010)^n(01001)^n0$

Example:



$$\begin{aligned} p(S_3) &= 0(10010)^3(01001)^30 \\ &= 01001010010100100100101001010010 \end{aligned}$$

\mathcal{L}_0 -structures

\mathcal{L}_0 -structures:

- their tiling sequences can contain only 2's (“go straight”)
- let S_n be a linear constructible structure described by the sequence: $\underbrace{2, 2, \dots, 2}_{n-1}$
- we have $p(S_n) = 0(10010)^n(01001)^n0$

Theorem 2: *For every $n \geq 1$, the protein $p(S_n)$ is stable, i.e., Conjecture 1 holds for all \mathcal{L}_0 -structures.*

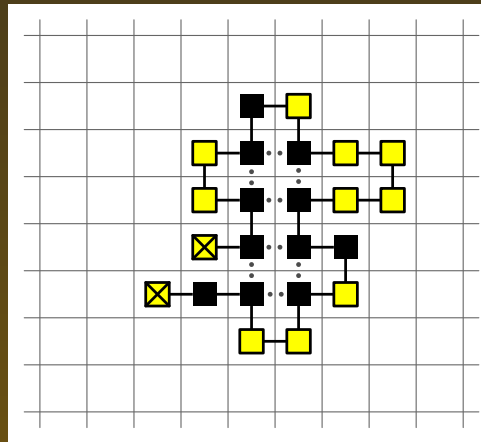
Diagonal frame

Diagonal frame — the smallest *diagonal* rectangle containing all hydrophobic “1” monomers.

Diagonal frame

Diagonal frame — the smallest *diagonal* rectangle containing all hydrophobic “1” monomers.

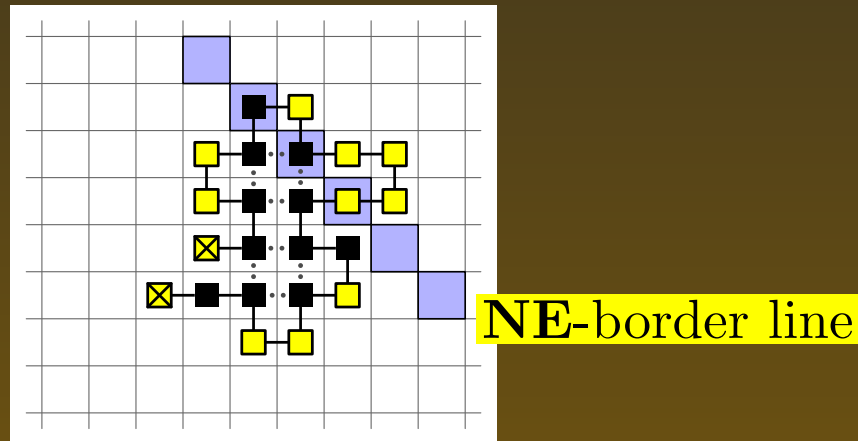
Example:



Diagonal frame

Diagonal frame — the smallest *diagonal* rectangle containing all hydrophobic “1” monomers.

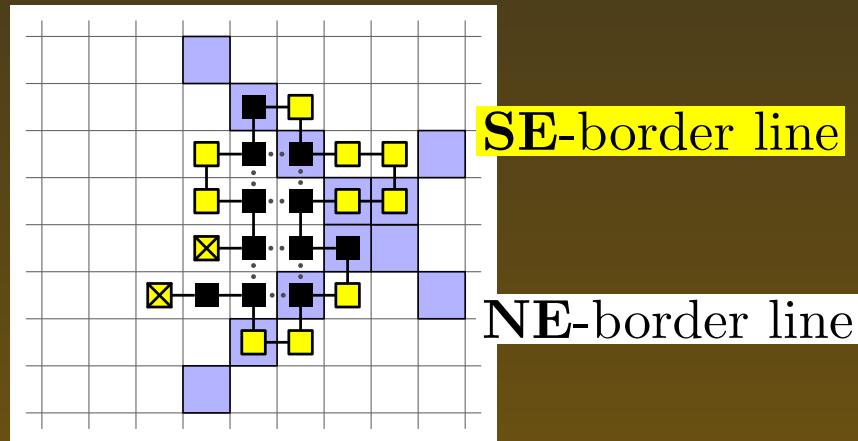
Example:



Diagonal frame

Diagonal frame — the smallest *diagonal* rectangle containing all hydrophobic “1” monomers.

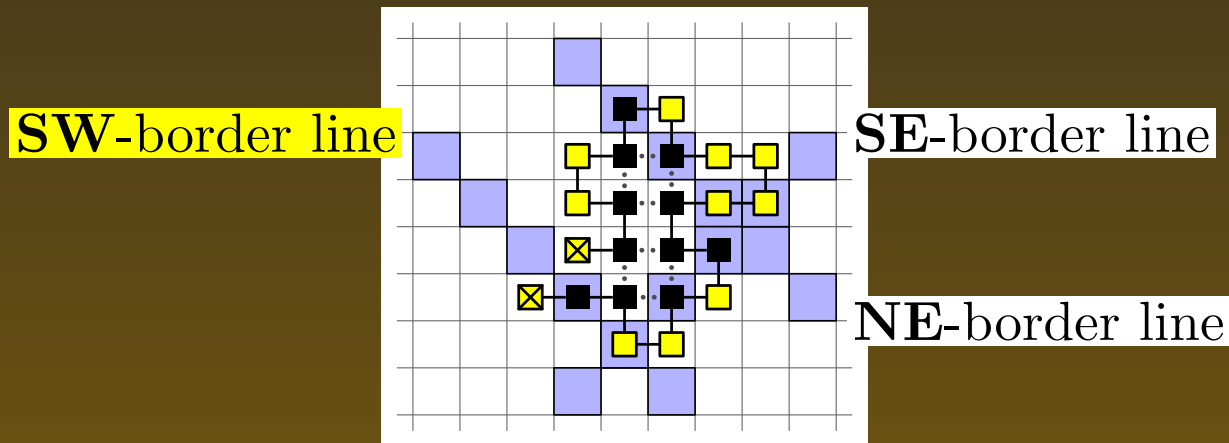
Example:



Diagonal frame

Diagonal frame — the smallest *diagonal* rectangle containing all hydrophobic “1” monomers.

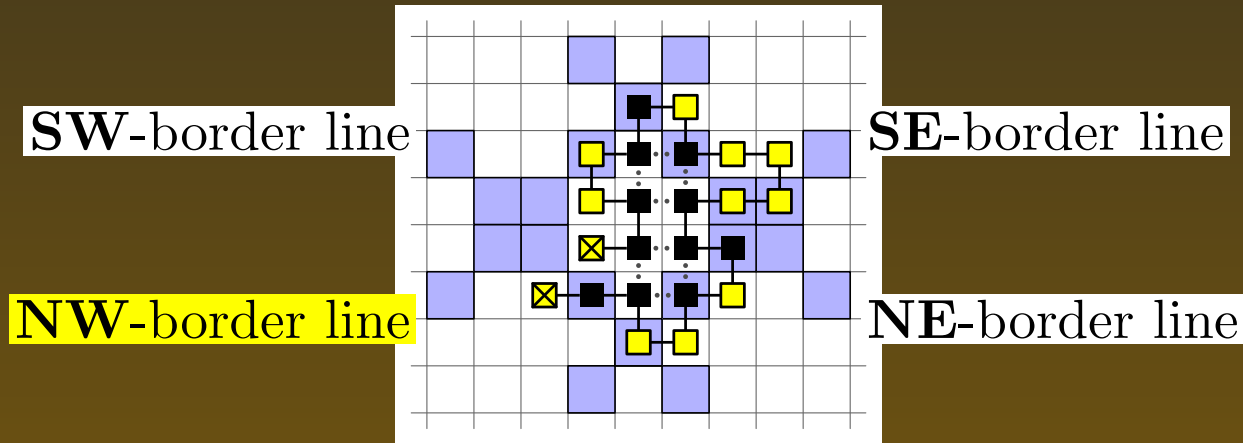
Example:



Diagonal frame

Diagonal frame — the smallest *diagonal* rectangle containing all hydrophobic “1” monomers.

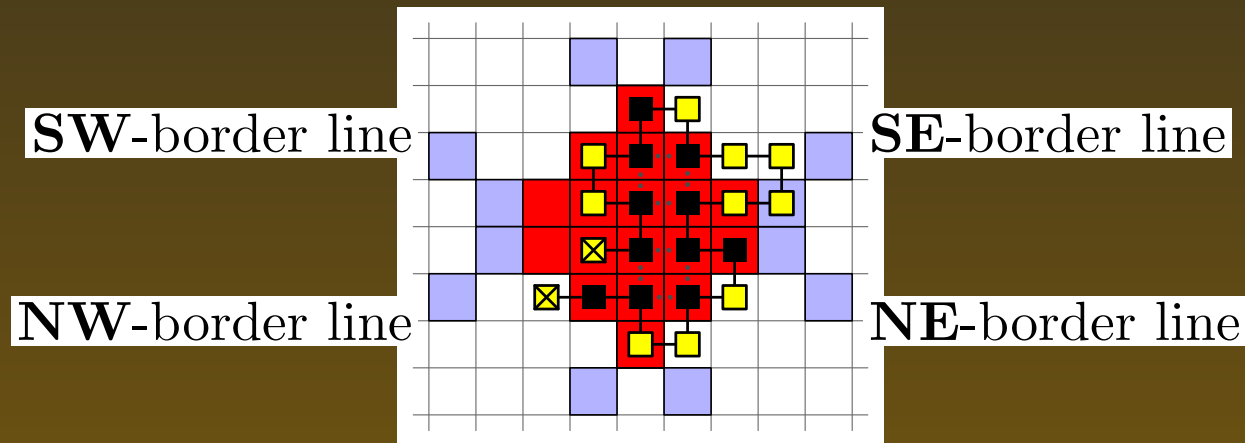
Example:



Diagonal frame

Diagonal frame — the smallest *diagonal* rectangle containing all hydrophobic “1” monomers.

Example:



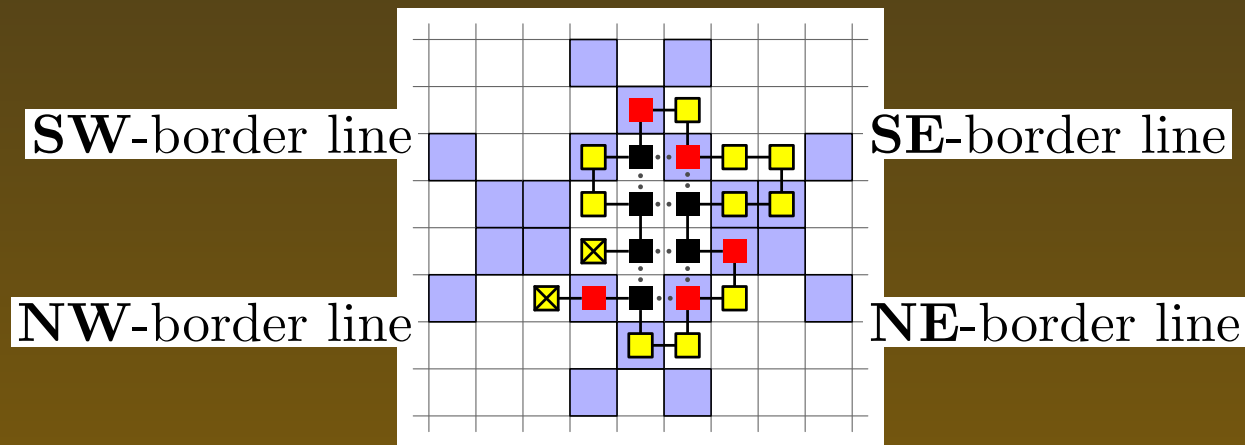
Note that all hydrophobic “1” monomers lie inside the **diagonal frame**, but some polar “0” monomers can lie outside.

Diagonal frame

Diagonal frame — the smallest *diagonal* rectangle containing all hydrophobic “1” monomers.

The hydrophobic “1” monomers lying on the border of the diagonal frame are called *boundary squares*.

Example:



In this example we have **5 boundary squares**.

Diagonal frame

The hydrophobic “1” monomers lying on the border of the diagonal frame are called *boundary squares*.

Observation:

If the fold is *saturated* there are at least *four* boundary squares, and at least *two* are not adjacent to a *terminal* (*terminal* = the first or the last monomer of the protein sequence).

Diagonal frame

The hydrophobic “1” monomers lying on the border of the diagonal frame are called *boundary squares*.

Observation:

If the fold is *saturated* there are at least *four* boundary squares, and at least *two* are not adjacent to a *terminal* (*terminal* = the first or the last monomer of the protein sequence).

- there cannot be a boundary square in the corner: by *Observation 3(a)*, every boundary square should be adjacent to two hydrophobic “1” monomers

Diagonal frame

The hydrophobic “1” monomers lying on the border of the diagonal frame are called *boundary squares*.

Observation:

If the fold is *saturated* there are at least *four* boundary squares, and at least *two* are not adjacent to a *terminal* (*terminal* = the first or the last monomer of the protein sequence).

- there cannot be a boundary square in the corner: by *Observation 3(a)*, every boundary square should be adjacent to two hydrophobic “1” monomers
- each border line should contain at least one boundary square (the diagonal frame is *minimal*)

Diagonal frame

Observation:

If the fold is *saturated* there are at least *four* boundary squares, and at least *two* are not adjacent to a *terminal* (*terminal* = the first or the last monomer of the protein sequence).

- there cannot be a boundary square in the corner: by *Observation 3(a)*, every boundary square should be adjacent to two hydrophobic “1” monomers
- each border line should contain at least one boundary square (the diagonal frame is *minimal*)
- hence, there are at least *four*

Diagonal frame

Observation:

If the fold is *saturated* there are at least *four* boundary squares, and at least *two* are not adjacent to a *terminal* (*terminal* = the first or the last monomer of the protein sequence).

- there cannot be a boundary square in the corner: by *Observation 3(a)*, every boundary square should be adjacent to two hydrophobic “1” monomers
- each border line should contain at least one boundary square (the diagonal frame is *minimal*)
- hence, there are at least *four*
- a terminal is adjacent to at most one boundary square

Diagonal frame

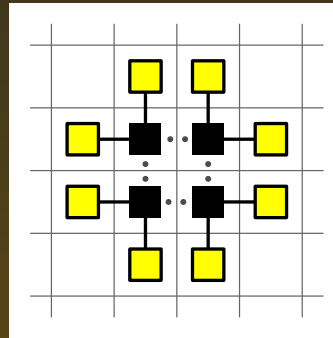
Observation:

If the fold is *saturated* there are at least *four* boundary squares, and at least *two* are not adjacent to a *terminal*.

- there cannot be a boundary square in the corner: by *Observation 3(a)*, every boundary square should be adjacent to two hydrophobic “1” monomers
- each border line should contain at least one boundary square (the diagonal frame is *minimal*)
- hence, there are at least *four*
- a terminal is adjacent to at most one boundary square
- hence, there are at least *two* non-adjacent to a terminal

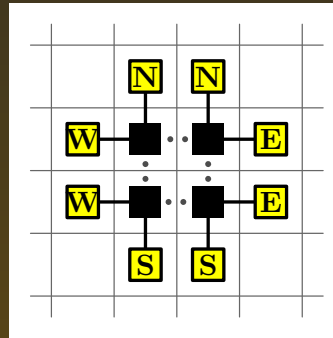
Cores

A *core* is the following configuration of monomers:



Cores

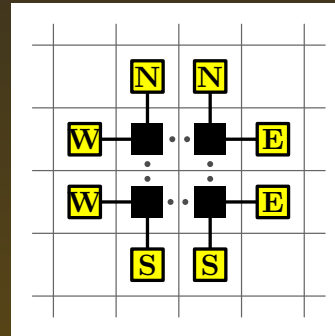
A *core* is the following configuration of monomers:



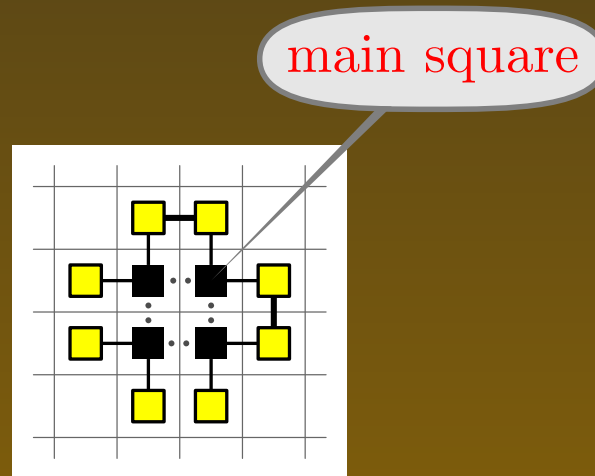
We will consider different type of cores:

Cores

A *core* is the following configuration of monomers:



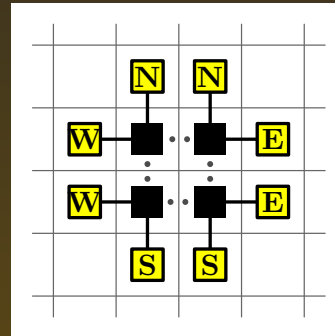
We will consider different type of cores:



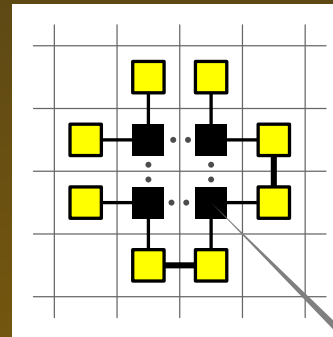
NE-closed core:

Cores

A *core* is the following configuration of monomers:



We will consider different type of cores:

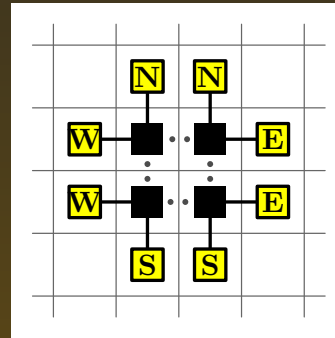


main square

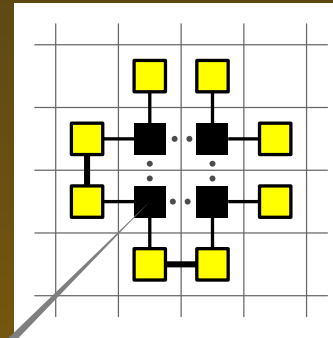
SE-closed core:

Cores

A *core* is the following configuration of monomers:



We will consider different type of cores:

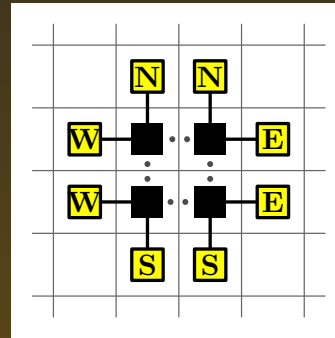


SW-closed core:

main square

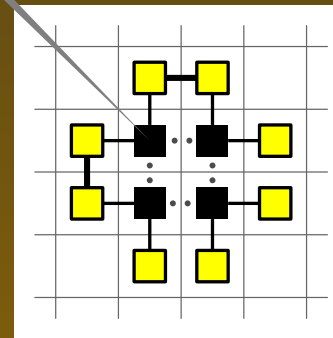
Cores

A *core* is the following configuration of monomers:



We will consider different type of cores:

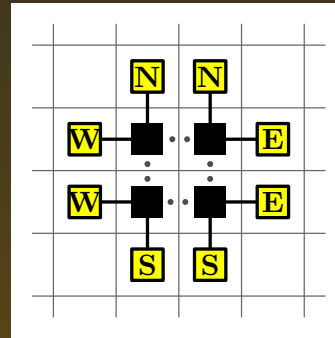
main square



NW-closed core:

Cores

A *core* is the following configuration of monomers:



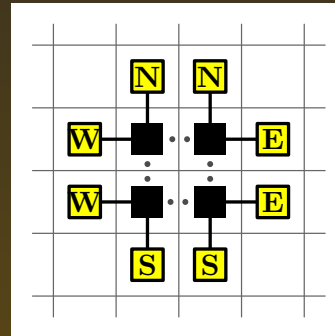
We will consider different type of cores:

NE-closed core
SE-closed core
SW-closed core
NW-closed core

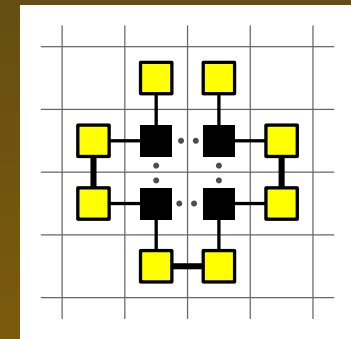
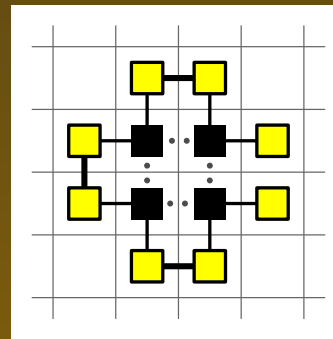
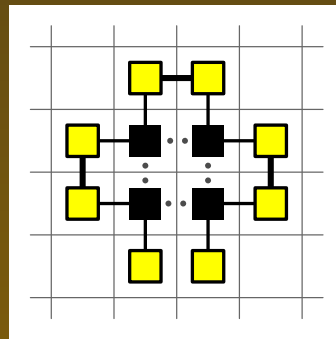
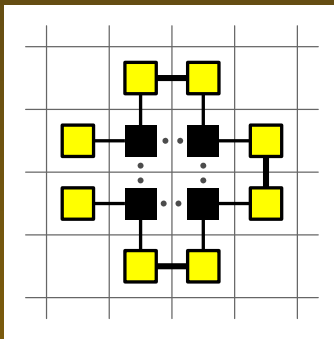
} *corner-closed cores*

Cores

A *core* is the following configuration of monomers:

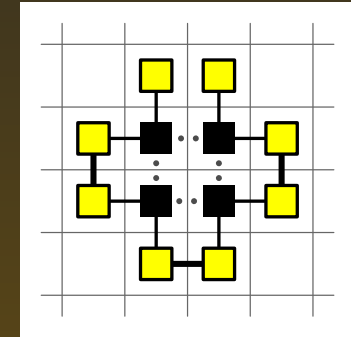
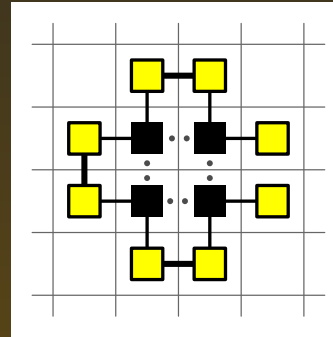
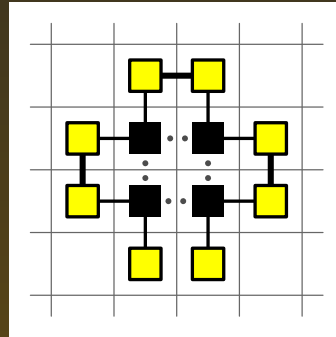
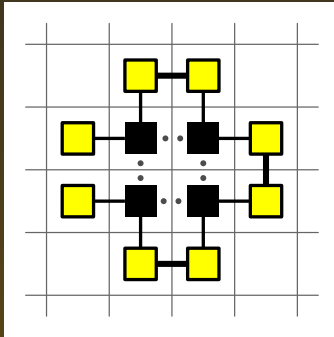


We will consider different type of cores:
completely-closed cores:



Cores

completely-closed cores:

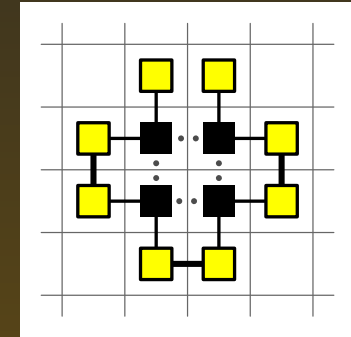
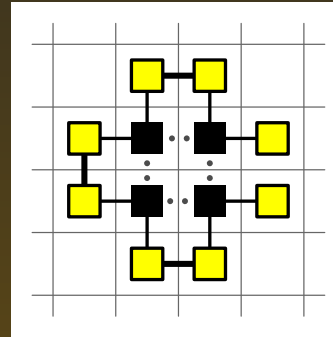
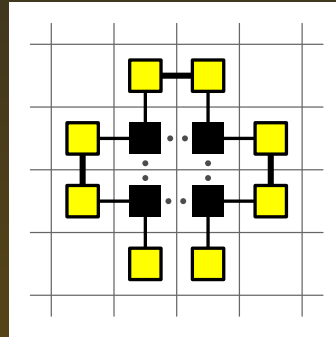
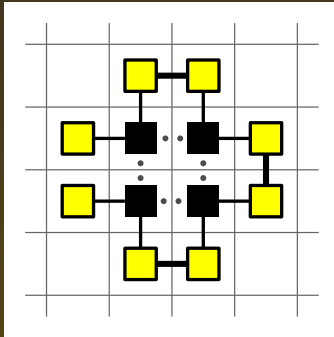


Note.

- For every linear constructible structure S , the fold $c(S)$ has *exactly one* completely-closed core.

Cores

completely-closed cores:

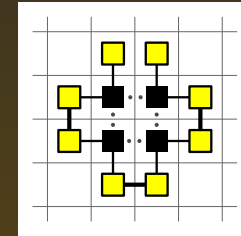
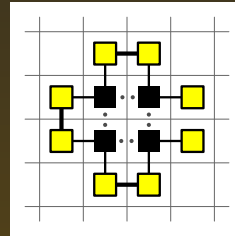
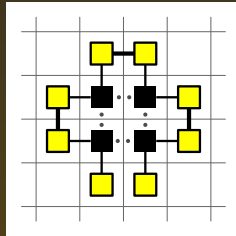
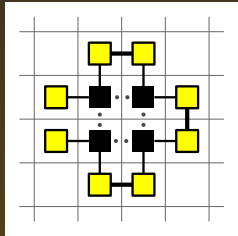


Note.

- For every linear constructible structure S , the fold $c(S)$ has *exactly one* completely-closed core.
- Our goal is to consider any saturated fold of $p(S)$ and identify a completely-closed core in it.

Cores

completely-closed cores:



Note.

- For every linear constructible structure S , the fold $c(S)$ has *exactly one* completely-closed core.
- Our goal is to consider any saturated fold of $p(S)$ and identify a completely-closed core in it.
- After cutting the core away we get a saturated fold of $p(S')$, where S' is the linear constructible structure obtained from S by removing one regular tile.

Note.

- For every linear constructible structure S , the fold $c(S)$ has *exactly one* completely-closed core.
- Our goal is to consider any saturated fold of $p(S)$ and identify a completely-closed core in it.
- After cutting the core away we get a saturated fold of $p(S')$, where S' is the linear constructible structure obtained from S by removing one regular tile.
- Then, we can apply induction to show that the chosen saturated fold is unique, i.e., equivalent to $c(S)$.

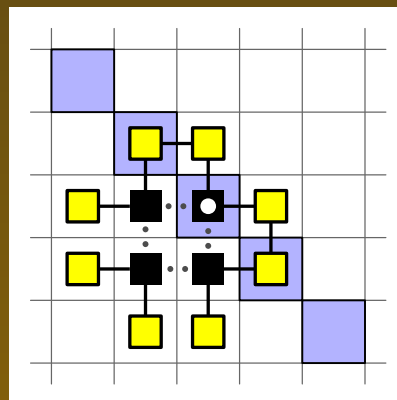
Lemma 2

- The boundary squares are useful to localize a completely-closed core.

Lemma 2

- The boundary squares are useful to localize a completely-closed core.

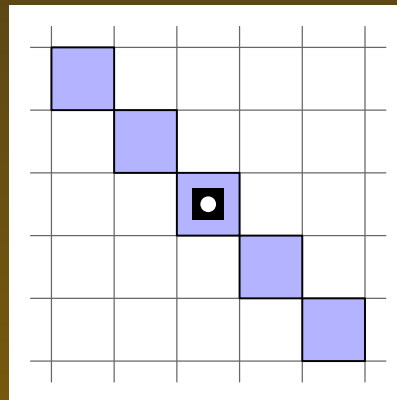
Lemma 2: *Let $p \in \{0, 1\}^*$ be a protein not containing 11, 000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\text{NE}, \text{SE}, \text{SW}, \text{NW}\}$, each boundary square s lying on the X -border line not adjacent to a terminal lying outside of the diagonal frame of the fold is the main square of a X -closed core.*



Lemma 2

Lemma 2: *Let $p \in 0\{0,1\}^*0$ be a protein not containing 11, 000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\mathbf{NE}, \mathbf{SE}, \mathbf{SW}, \mathbf{NW}\}$, each boundary square s lying on the X -border line not adjacent to a terminal lying outside of the diagonal frame of the fold is the main square of a X -closed core.*

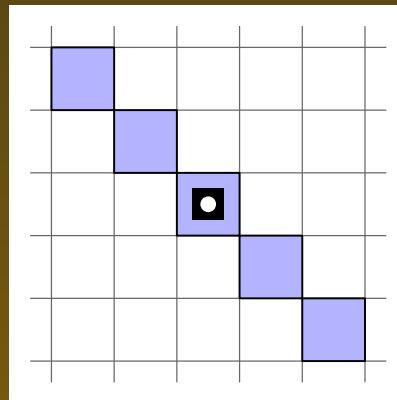
Proof. Consider a boundary square lying on **NE**-border line:



Lemma 2

Lemma 2: *Let $p \in 0\{0,1\}^*0$ be a protein not containing 11, 000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\mathbf{NE}, \mathbf{SE}, \mathbf{SW}, \mathbf{NW}\}$, each boundary square s lying on the X -border line not adjacent to a terminal lying outside of the diagonal frame of the fold is the main square of a X -closed core.*

Proof. Consider a boundary square lying on **NE**-border line:

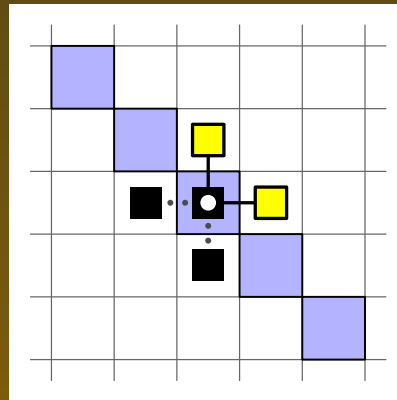


By *Observation 3(a)*, it has two neighboring 0-monomers and two neighboring 1-monomers.

Lemma 2

Lemma 2: *Let $p \in 0\{0,1\}^*0$ be a protein not containing 11, 000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\mathbf{NE}, \mathbf{SE}, \mathbf{SW}, \mathbf{NW}\}$, each boundary square s lying on the X -border line not adjacent to a terminal lying outside of the diagonal frame of the fold is the main square of a X -closed core.*

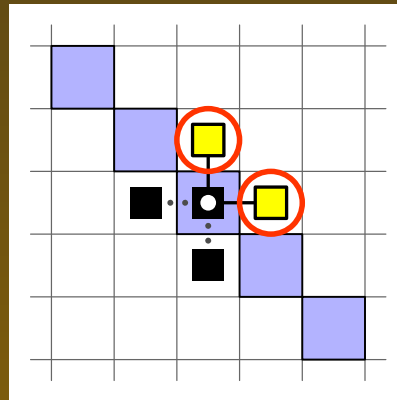
Proof. There are no 1-monomers outside the diagonal frame, hence we have the following configuration:



Lemma 2

Lemma 2: *Let $p \in 0\{0,1\}^*0$ be a protein not containing 11, 000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\mathbf{NE}, \mathbf{SE}, \mathbf{SW}, \mathbf{NW}\}$, each boundary square s lying on the X -border line **not adjacent to a terminal** lying outside of the diagonal frame of the fold is the main square of a X -closed core.*

Proof. There are no 1-monomers outside the diagonal frame, hence we have the following configuration:

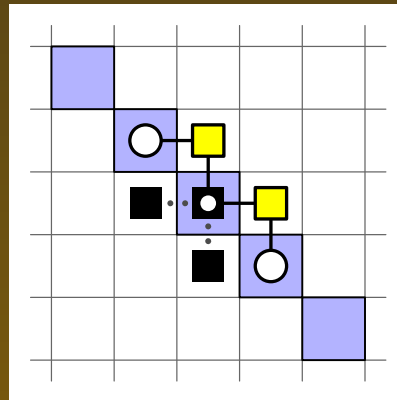


The 0-monomers cannot be terminals.

Lemma 2

Lemma 2: *Let $p \in 0\{0,1\}^*0$ be a protein not containing 11, 000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\mathbf{NE}, \mathbf{SE}, \mathbf{SW}, \mathbf{NW}\}$, each boundary square s lying on the X -border line not adjacent to a terminal lying outside of the diagonal frame of the fold is the main square of a X -closed core.*

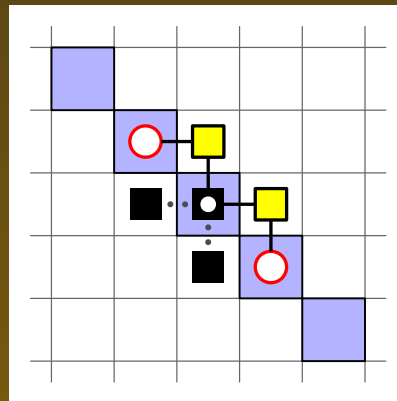
Proof. They have to connect to squares inside the frame.



Lemma 2

Lemma 2: *Let $p \in 0\{0,1\}^*0$ be a protein not containing 11, 000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\mathbf{NE}, \mathbf{SE}, \mathbf{SW}, \mathbf{NW}\}$, each boundary square s lying on the X -border line not adjacent to a terminal lying outside of the diagonal frame of the fold is the main square of a X -closed core.*

Proof. They have to connect to squares inside the frame.

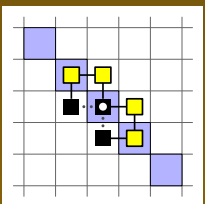
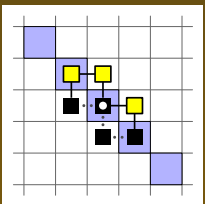
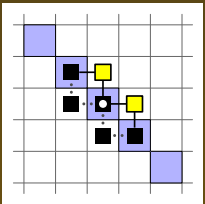


We have three possibilities (up to symmetry) what type the red monomers are.

Lemma 2

Lemma 2: *Let $p \in 0\{0,1\}^*0$ be a protein not containing 11, 000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\mathbf{NE}, \mathbf{SE}, \mathbf{SW}, \mathbf{NW}\}$, each boundary square s lying on the X -border line not adjacent to a terminal lying outside of the diagonal frame of the fold is the main square of a X -closed core.*

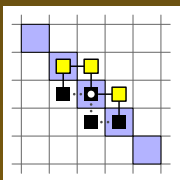
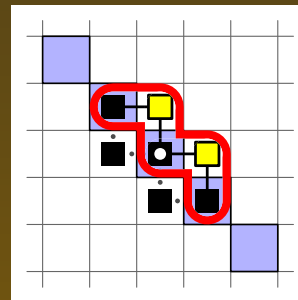
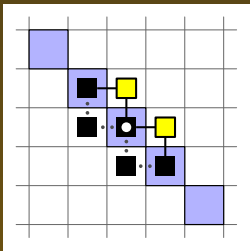
Proof.



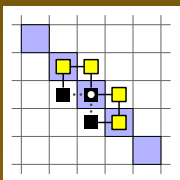
Lemma 2

Lemma 2: *Let $p \in 0\{0,1\}^*0$ be a protein not containing 11, 000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\mathbf{NE}, \mathbf{SE}, \mathbf{SW}, \mathbf{NW}\}$, each boundary square s lying on the X -border line not adjacent to a terminal lying outside of the diagonal frame of the fold is the main square of a X -closed core.*

Proof. Case 1



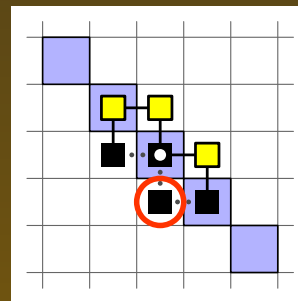
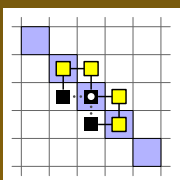
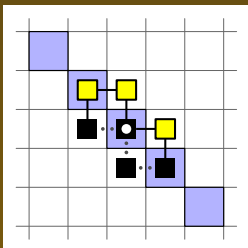
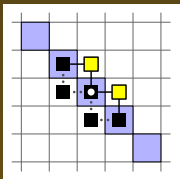
A contradiction: the protein sequence does not contain substring 10101.



Lemma 2

Lemma 2: *Let $p \in 0\{0,1\}^*0$ be a protein not containing 11, 000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\mathbf{NE}, \mathbf{SE}, \mathbf{SW}, \mathbf{NW}\}$, each boundary square s lying on the X -border line not adjacent to a terminal lying outside of the diagonal frame of the fold is the main square of a X -closed core.*

Proof. Case 2

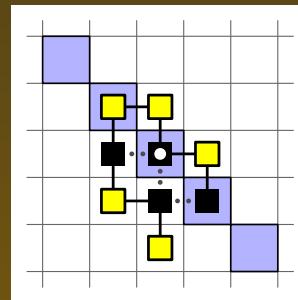
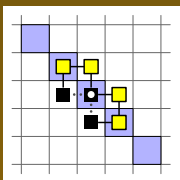
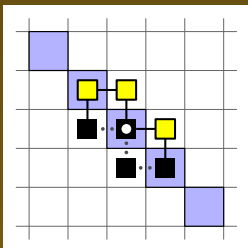
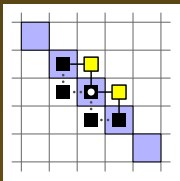


The circled **1-monomer** has already two neighboring **1-monomers**, hence the other two neighbors are **0-monomer**.

Lemma 2

Lemma 2: *Let $p \in \{0,1\}^*$ be a protein not containing 11, 000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\mathbf{NE}, \mathbf{SE}, \mathbf{SW}, \mathbf{NW}\}$, each boundary square s lying on the X -border line not adjacent to a terminal lying outside of the diagonal frame of the fold is the main square of a X -closed core.*

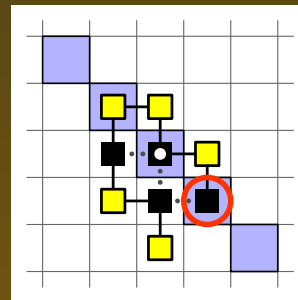
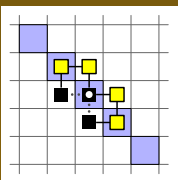
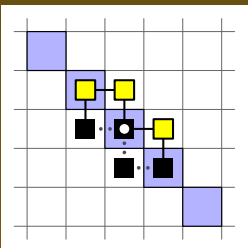
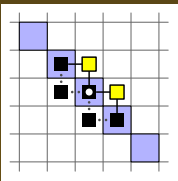
Proof. Case 2



Lemma 2

Lemma 2: *Let $p \in \{0,1\}^*0$ be a protein not containing 11, 000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\mathbf{NE}, \mathbf{SE}, \mathbf{SW}, \mathbf{NW}\}$, each boundary square s lying on the X -border line not adjacent to a terminal lying outside of the diagonal frame of the fold is the main square of a X -closed core.*

Proof. Case 2

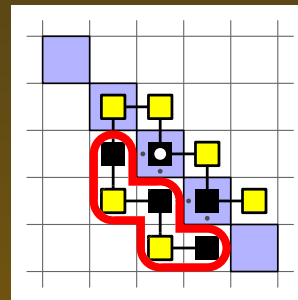
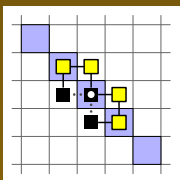
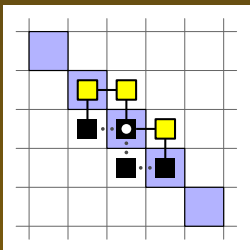
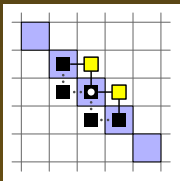


The eastern neighbor of the circled 1-monomer lies outside of the frame. Hence, it must be 0-monomer, and the southern neighbor is 1-monomer.

Lemma 2

Lemma 2: Let $p \in 0\{0,1\}^*0$ be a protein *not containing* 11, 000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\mathbf{NE}, \mathbf{SE}, \mathbf{SW}, \mathbf{NW}\}$, each boundary square s lying on the X -border line not adjacent to a terminal lying outside of the diagonal frame of the fold is the main square of a X -closed core.

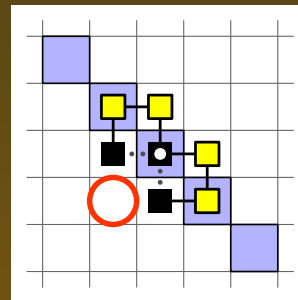
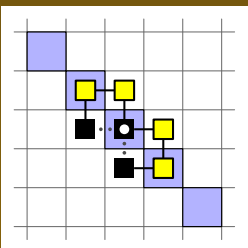
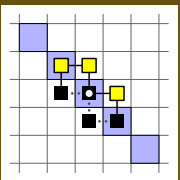
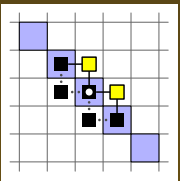
Proof. Case 2



Lemma 2

Lemma 2: *Let $p \in 0\{0,1\}^*0$ be a protein not containing 11, 000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\mathbf{NE}, \mathbf{SE}, \mathbf{SW}, \mathbf{NW}\}$, each boundary square s lying on the X -border line not adjacent to a terminal lying outside of the diagonal frame of the fold is the main square of a X -closed core.*

Proof. Case 3

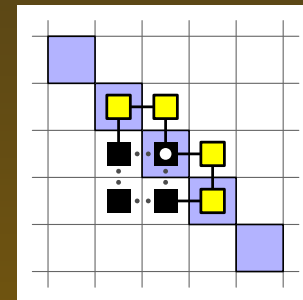
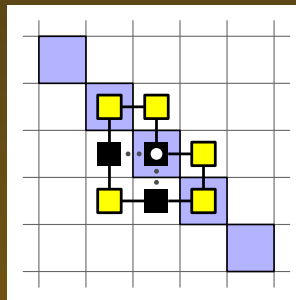
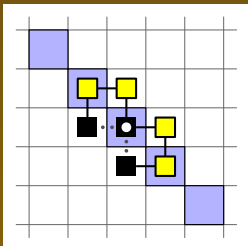
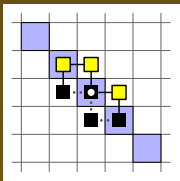
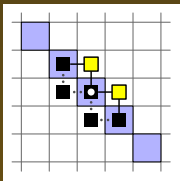


Consider two cases depending on the type of monomer in the circled **square**.

Lemma 2

Lemma 2: *Let $p \in 0\{0,1\}^*0$ be a protein not containing 11, 000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\mathbf{NE}, \mathbf{SE}, \mathbf{SW}, \mathbf{NW}\}$, each boundary square s lying on the X -border line not adjacent to a terminal lying outside of the diagonal frame of the fold is the main square of a X -closed core.*

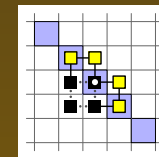
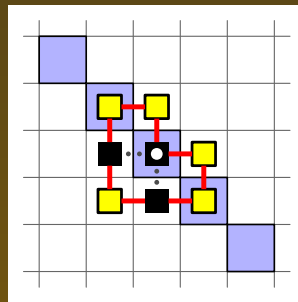
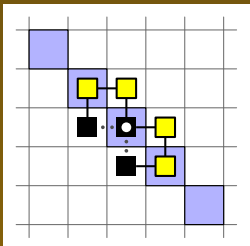
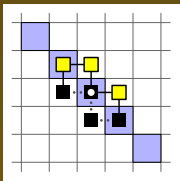
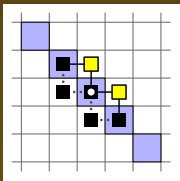
Proof. Case 3



Lemma 2

Lemma 2: *Let $p \in 0\{0,1\}^*0$ be a protein not containing 11, 000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\mathbf{NE}, \mathbf{SE}, \mathbf{SW}, \mathbf{NW}\}$, each boundary square s lying on the X -border line not adjacent to a terminal lying outside of the diagonal frame of the fold is the main square of a X -closed core.*

Proof. Case 3

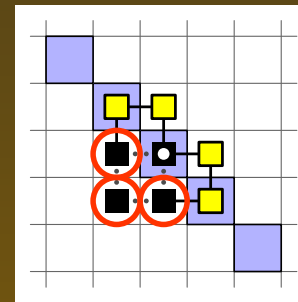
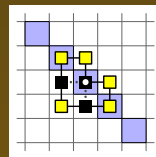
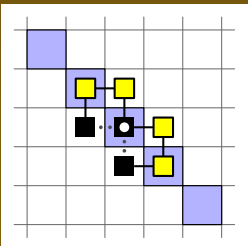
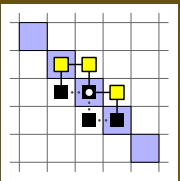
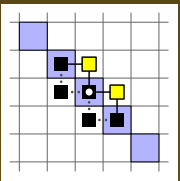


In the first case, we have a **closed sequence** of monomers in the fold, a contradiction.

Lemma 2

Lemma 2: *Let $p \in 0\{0,1\}^*0$ be a protein not containing 11, 000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\mathbf{NE}, \mathbf{SE}, \mathbf{SW}, \mathbf{NW}\}$, each boundary square s lying on the X -border line not adjacent to a terminal lying outside of the diagonal frame of the fold is the main square of a X -closed core.*

Proof. Case 3

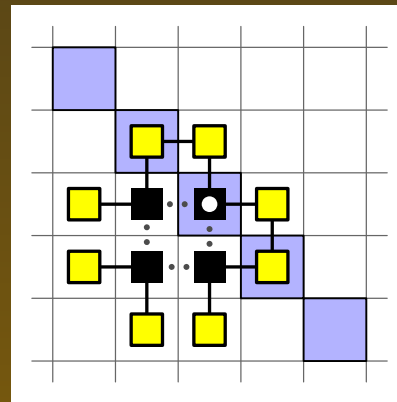
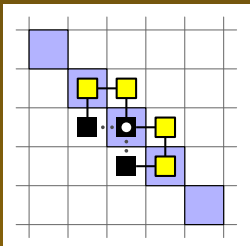
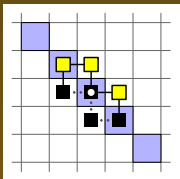
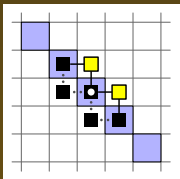


In the second case, the circled **1-monomers** have already two neighboring **1-monomers**. Hence, they remaining neighbors are all **0-monomers**.

Lemma 2

Lemma 2: *Let $p \in 0\{0,1\}^*0$ be a protein not containing 11,000 and 10101 as a substring. For every saturated fold of p and every $X \in \{\mathbf{NE}, \mathbf{SE}, \mathbf{SW}, \mathbf{NW}\}$, each boundary square s lying on the X -border line not adjacent to a terminal lying outside of the diagonal frame of the fold is the main square of a X -closed core.*

Proof. Case 3



We have a **NE**-closed core. Done.

Proof of Theorem 2

Theorem 2: *For every $n \geq 1$, the protein $p(S_n) = 0(10010)^n(01001)^n0$ is stable.*

Proof of Theorem 2

Theorem 2: *For every $n \geq 1$, the protein $p(S_n) = 0(10010)^n(01001)^n0$ is stable.*

Proof.

- Note that $p(S_n)$ does not contain 10101 as a substring, hence we can apply *Lemma 2*.

Proof of Theorem 2

Theorem 2: *For every $n \geq 1$, the protein $p(S_n) = 0(10010)^n(01001)^n0$ is stable.*

Proof.

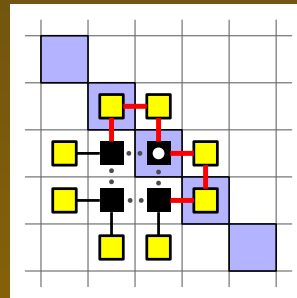
- Note that $p(S_n)$ does not contain 10101 as a substring, hence we can apply *Lemma 2*.
- We have at least two boundary squares, say s and t , not adjacent to a terminal. By *Lemma 2*, they are main squares of corner-closed cores.

Proof of Theorem 2

Theorem 2: *For every $n \geq 1$, the protein $p(S_n)$ is stable.*

Proof.

- Note that $p(S_n)$ does not contain 10101 as a substring, hence we can apply *Lemma 2*.
- We have at least two boundary squares, say s and t , not adjacent to a terminal. By *Lemma 2*, they are main squares of corner-closed cores.
- Note that each *corner-closed core* contains a sequence 1001001 of consecutive monomers.



Proof of Theorem 2

Theorem 2: *For every $n \geq 1$, the protein $p(S_n)$ is stable.*

- Note that $p(S_n)$ does not contain **10101** as a substring, hence we can apply *Lemma 2*.
- We have at least two boundary squares, say s and t , not adjacent to a terminal. By *Lemma 2*, they are main squares of corner-closed cores.
- Note that each *corner-closed core* contains a sequence **1001001** of consecutive monomers.
- There are exactly *two* occurrences of the substring **1001001** in the protein sequence $p(S_n) = 0(10010)^n(01001)^n0$ (in the middle), and they *overlap*.

Proof of Theorem 2

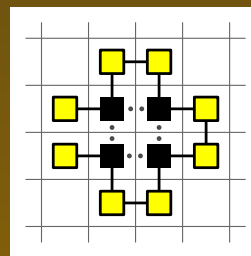
Theorem 2: *For every $n \geq 1$, the protein $p(S_n)$ is stable.*

- We have at least two boundary squares, say s and t , not adjacent to a terminal. By *Lemma 2*, they are main squares of corner-closed cores.
- Note that each *corner-closed core* contains a sequence 1001001 of consecutive monomers.
- There are exactly *two* occurrences of the substring 1001001 in the protein sequence $p(S_n) = 0(10010)^n(01001)^n0$ (in the middle), and they *overlap*.
- Hence, s and t must be main squares of the same *completely-closed core*.

Proof of Theorem 2

Theorem 2: *For every $n \geq 1$, the protein $p(S_n)$ is stable.*

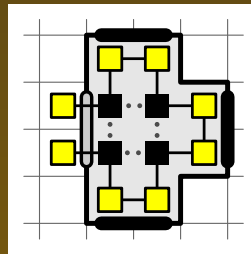
- Note that each *corner-closed core* contains a sequence 1001001 of consecutive monomers.
- There are exactly *two* occurrences of the substring 1001001 in the protein sequence $p(S_n) = 0(10010)^n(01001)^n0$ (in the middle), and they *overlap*.
- Hence, s and t must be main squares of the same *completely-closed core*.



Proof of Theorem 2

Theorem 2: *For every $n \geq 1$, the protein $p(S_n)$ is stable.*

- There are exactly *two* occurrences of the substring 1001001 in the protein sequence $p(S_n) = 0(10010)^n(01001)^n0$ (in the middle), and they *overlap*.
- Hence, s and t must be main squares of the same *completely-closed* core.



- We have a regular tile. If we cut it away we get a saturated fold of the protein $p(S_{n-1})$ and we can apply induction.

Proof of Theorem 2

Theorem 2: *For every $n \geq 1$, the protein $p(S_n)$ is stable.*

- There are exactly *two* occurrences of the substring 1001001 in the protein sequence $p(S_n) = 0(10010)^n(01001)^n0$ (in the middle), and they *overlap*.
- Hence, s and t must be main squares of the same *completely-closed* core.
- We have a regular tile. If we cut it away we get a saturated fold of the protein $p(S_{n-1})$ and we can apply induction.
- Done.

\mathcal{L}_1 -structures

\mathcal{L}_1 -structures:

- their tiling sequences contains exactly one “bend” (either 1 or 3) and the rest are 2’s (“go straight”)

\mathcal{L}_1 -structures

\mathcal{L}_1 -structures:

- their tiling sequences contains exactly one “bend” (either 1 or 3) and the rest are 2’s (“go straight”)
- let $L_{n,m}$ be a linear constructible structure described by the tiling sequence: $\underbrace{2, 2, \dots, 2}_{n-1}, 3, \underbrace{2, 2, \dots, 2}_{m-1}$

\mathcal{L}_1 -structures

\mathcal{L}_1 -structures:

■ their tiling sequences contains exactly one “bend” (either 1 or 3) and the rest are 2’s (“go straight”)

■ let $L_{n,m}$ be a linear constructible structure described by the tiling sequence:

$$\underbrace{2, 2, \dots, 2}_{n-1}, 3, \underbrace{2, 2, \dots, 2}_{m-1}$$

■ we have

$$p(L_{n,m}) = 0(10010)^n 010(10010)^m (01001)^m 01(01001)^{n-1} 0$$

\mathcal{L}_1 -structures

- let $L_{n,m}$ be a linear constructible structure described by the tiling sequence:

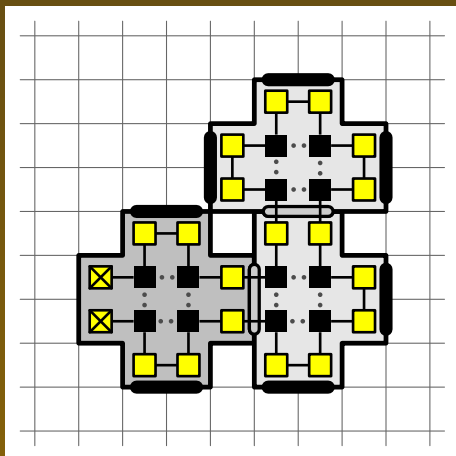
$$\underbrace{2, 2, \dots, 2}_{n-1}, 3, \underbrace{2, 2, \dots, 2}_{m-1}$$

- we have

$$p(L_{n,m}) = 0(10010)^n 010(10010)^m (01001)^m 01(01001)^{n-1} 0$$

Example: $L_{2,1}$ — tiling sequence 2, 3

$$p(L_{2,1}) = 01001010010010100100100101010010$$



\mathcal{L}_1 -structures

- let $L_{n,m}$ be a linear constructible structure described by the tiling sequence:

$$\underbrace{2, 2, \dots, 2}_{n-1}, 3, \underbrace{2, 2, \dots, 2}_{m-1}$$

- we have

$$p(L_{n,m}) = 0(10010)^n 010(10010)^m (01001)^m 01(01001)^{n-1} 0$$

Theorem 3: *For every $n, m \geq 1$, the protein $p(L_{n,m})$ is stable. Hence, by symmetry, Conjecture 1 holds for all \mathcal{L}_1 -structures.*

\mathcal{L}_1 -structures

Theorem 3: *For every $n, m \geq 1$, the protein $p(L_{n,m})$ is stable. Hence, by symmetry, Conjecture 1 holds for all \mathcal{L}_1 -structures.*

Remarks on the proof.

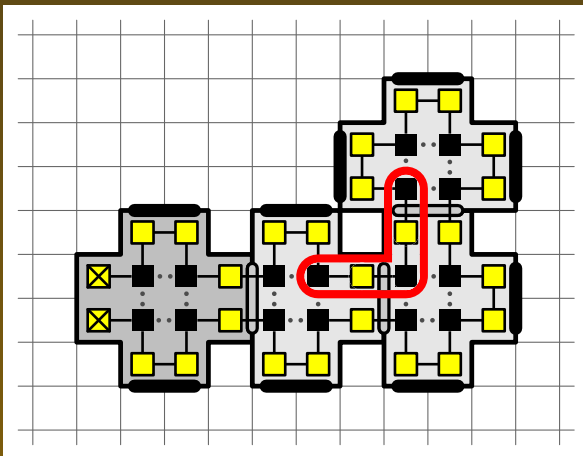
\mathcal{L}_1 -structures

Theorem 3: *For every $n, m \geq 1$, the protein $p(L_{n,m})$ is stable.*

Remarks on the proof. The protein sequence $p(L_{n,m})$ contains **one occurrence of the substring “10101”**.

Example: $L_{3,1}$ — tiling sequence 2, 2, 3

$p(L_{3,1}) = 01001010010 \ 10010010100100100101001010010010$



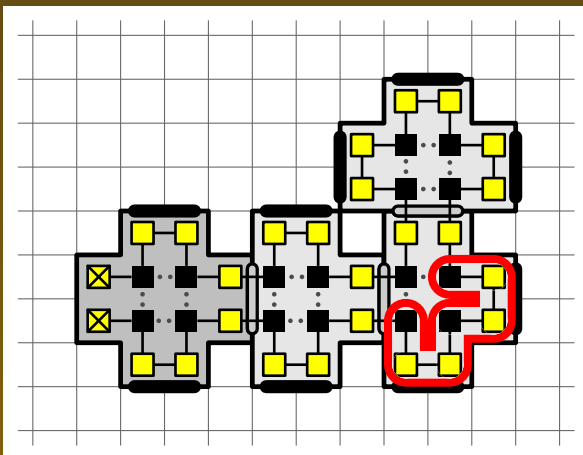
\mathcal{L}_1 -structures

Theorem 3: *For every $n, m \geq 1$, the protein $p(L_{n,m})$ is stable.*

Remarks on the proof. The protein sequence $p(L_{n,m})$ contains one occurrence of the substring “10101” and **three occurrences of the substring “1001001001”**.

Example: $L_{3,1}$ — tiling sequence 2, 2, 3

$p(L_{3,1}) = 01001010010$ **1001001** 010010010010100101010010



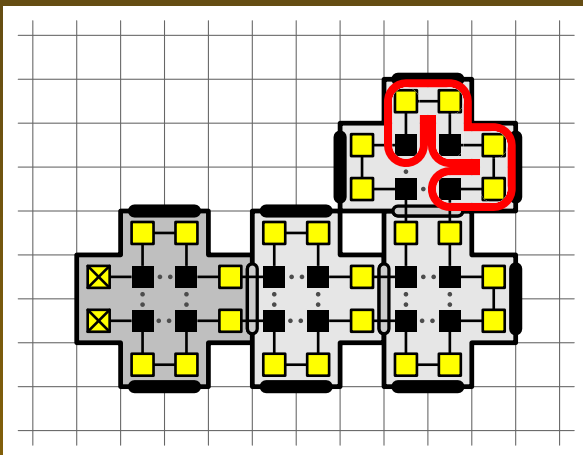
\mathcal{L}_1 -structures

Theorem 3: *For every $n, m \geq 1$, the protein $p(L_{n,m})$ is stable.*

Remarks on the proof. The protein sequence $p(L_{n,m})$ contains one occurrence of the substring “10101” and **three occurrences of the substring “1001001001”**.

Example: $L_{3,1}$ — tiling sequence 2, 2, 3

$p(L_{3,1}) = 01001010010 \ 1001001010010010010100101010010$



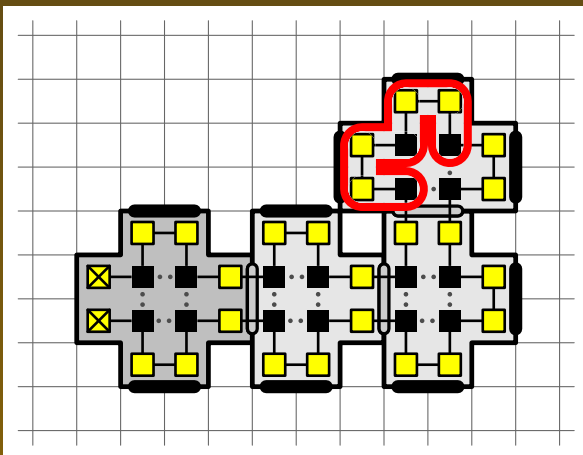
\mathcal{L}_1 -structures

Theorem 3: *For every $n, m \geq 1$, the protein $p(L_{n,m})$ is stable.*

Remarks on the proof. The protein sequence $p(L_{n,m})$ contains one occurrence of the substring “10101” and **three occurrences of the substring “1001001001”**.

Example: $L_{3,1}$ — tiling sequence 2, 2, 3

$p(L_{3,1}) = 01001010010 \ 1001001010010010010100101010010$



\mathcal{L}_1 -structures

Theorem 3: *For every $n, m \geq 1$, the protein $p(L_{n,m})$ is stable.*

Remarks on the proof. The protein sequence $p(L_{n,m})$ contains one occurrence of the substring “10101” and three occurrences of the substring “1001001001”.

Hence, the *Lemma 2* cannot be directly applied.

The proof is more technical and involves a lengthy analysis.

Conclusions

Future work:

- ▣ Prove Conjecture 2, or even Conjecture 1.

Conclusions

Future work:

- Prove Conjecture 2, or even Conjecture 1.
- Show similar results for other than square lattices (triangular, hexagonal, etc.).

Conclusions

Future work:

- Prove Conjecture 2, or even Conjecture 1.
- Show similar results for other than square lattices (triangular, hexagonal, etc.).
- Show similar results in 3D.